

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática



IMPLEMENTAÇÃO DE NOVAS FUNCIONALIDADES
NA APLICAÇÃO AVALIAÇÃO DE DESEMPENHO
360° (SIGAV360) E CRIAÇÃO DO MÓDULO CRM
(SIGCRM)

projecto realizado na

Capgemini Portugal

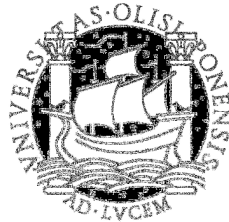
por

Sílvia Mendes Inácio

Mestrado em Engenharia Informática

2007

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática



IMPLEMENTAÇÃO DE NOVAS FUNCIONALIDADES
NA APLICAÇÃO AVALIAÇÃO DE DESEMPENHO
360° (SIGAV360) E CRIAÇÃO DO MÓDULO CRM
(SIGCRM)

projecto realizado na

Capgemini Portugal

por

Sílvia Mendes Inácio

Projecto orientado pelo Prof. Dr. Luís Antunes

e co-orientado por Fernando Amaral

Mestrado em Engenharia Informática

2007

Declaração

Sílvia Mendes Inácio, aluno nº 30364 da Faculdade de Ciências da Universidade de Lisboa, declara ceder os seus direitos de cópia sobre o seu Relatório de Projecto em Engenharia Informática, intitulado "Implementação de novas funcionalidades na aplicação Avaliação de Desempenho 360° (SigAV360) e criação do módulo CRM (SigCRM)", realizado no ano lectivo de 2006/2007 à Faculdade de Ciências da Universidade de Lisboa para o efeito de arquivo e consulta nas suas bibliotecas e publicação do mesmo em formato electrónico na Internet.

FCUL, 16 de Setembro de 2007

Fernando Amaral, supervisor do projecto de Sílvia Mendes Inácio, aluno da Faculdade de Ciências da Universidade de Lisboa, declara concordar com a divulgação do Relatório do Projecto em Engenharia Informática, intitulado "Implementação de novas funcionalidades na aplicação Avaliação de Desempenho 360° (SigAV360) e criação do módulo CRM (SigCRM)".

Local, 16 de Setembro de 2007

Resumo

Este documento descreve o trabalho realizado no âmbito da disciplina Projecto em Engenharia Informática (PEI) do Mestrado em Engenharia Informática (MEI) na Faculdade de Ciências da Universidade de Lisboa.

O projecto é desenvolvido numa instituição externa à faculdade, a Capgemini em Portugal, teve uma duração de 9 meses e foi inserido num estágio na instituição de acolhimento, no Departamento de Aplicações Standard.

Para este projecto foram planeados trabalhos em duas aplicações do departamento, num sistema de avaliações de desempenho 360°, intitulado Sistema Integrado de Gestão – Avaliação de Desempenho 360° (SigAV360) e num módulo de CRM (*Customer Relationship Management*), designado Sistema Integrado de Gestão – *Customer Relationship Management* (SigCRM). São ambas as aplicações em ambiente *Web*.

A aplicação SigAV360 já estava desenvolvida para um cliente específico do departamento, existindo a necessidade de tornar a aplicação *standard*.

Os objectivos a realizar nesta aplicação são de reestruturar, melhorar e implementar novas funcionalidades. Para isso seria necessário melhorar o aspecto visual da aplicação, modificar o menu de acesso às funcionalidades da aplicação, desenvolver um sistema de navegação autónomo, independente do *browser*, a par da inclusão de pontos de referência relativos às páginas de navegação e incluir algumas novas funcionalidades como a importação de empregados de uma empresa do módulo SigGP (Sistema Integrado de Gestão – Gestão de Pessoal), a consulta dos relatórios estatísticos das avaliações e a duplicação de avaliações.

Um levantamento de requisitos adequado e completo para uma aplicação *standard* de avaliações de desempenho, teria evitado esta reestruturação. Como contrapartida foi necessário realizar este levantamento de requisitos antes de se efectuar qualquer alteração à aplicação.

Com este levantamento de requisitos adequado foi possível reestruturar o modelo de dados e código desenvolvido para a aplicação, alcançando desta forma os objectivos definidos.

Para a aplicação SigCRM os objectivos são de criar o módulo CRM para o departamento, integrado numa equipa de desenvolvimento. Pretende-se que a equipa implemente na aplicação os conceitos de CRM.

As reuniões diárias de toda a equipa e uma boa gestão e planeamento do projecto permitiram que os objectivos iniciais fossem cumpridos.

Como base do desenvolvimento da aplicação usou-se algumas *Framework's* pertencentes ao Departamento das Aplicações Standard. Estas foram desenvolvidas com linguagens C#.NET e ASP.NET e permitiram um desenvolvimento rápido e estruturado da aplicação.

Ambas as aplicações foram desenvolvidas com as linguagens ASP.NET e C#.NET com o apoio da ferramenta Microsoft Visual Studio e Microsoft SQL Server.

PALAVRAS-CHAVE:

CRM (*Customer Relationship Management*), Avaliação de Desempenho 360°, C#.NET, ASP.NET, *Web*

Abstract

This document describes the work carried through in the scope of the class Projecto em Engenharia Informática (PEI) of the Mestrado em Engenharia Informática (MEI) in Faculdade de Ciências, Universidade de Lisboa.

The PEI was developed in an external institution to Faculdade de Ciências, in Capgemini at Portugal, it had a duration of 9 months and was inserted in a period of training in the institution, more precisely in the Department of Standard Applications.

For this project it has been planed to work in two applications of the department, first in a 360° system of performance evaluations, entitled Sistema Integrado de Gestão – Avaliação de Desempenho 360° (SigAV360) and then in a module of CRM (Customer Relationship Management), entitled Sistema Integrado de Gestão - Customer Relationship Management (SigCRM). Both of the applications are developed for a Web environment.

The SigAV360 application had been already developed for a specific customer of the department, but there is a need to make it a standard application.

The aims on this application are to restructure, improve and implement new features. To accomplish this, it will be necessary to improve the visual aspect of the application, modify the functionality access menu, develop a navigation system independent of the web browser, along with the inclusion of reference points on the navigation pages and include some new features such as the importation of employees of the company integrated at the SigGP (Sistema Integrado de Gestão – Gestão Pessoal) module, the access to statistical reports of the evaluations and the duplication of the data related to a process of evaluation.

A proper and complete removal of the requirements for a standard application of performance evaluations, could have avoided this necessary restructuring. In return, it was necessary to do this removal of requirements before any amendment on the application.

With this proper removal of requirements it was possible to restructure the data base model and the source code related to the application, accomplishing the initial objectives.

For the SigCRM application the goals were to create a module of CRM for the department, integrated in a team of development. The aim is to develop the CRM concepts.

The daily meetings of the entire team and a good management and planning of the project permitted to accomplish this aim.

Some Framework's of the Department of Standard Applications were used as a base structure of the project. They have been developed using C#.NET and ASP.NET languages and enabled a rapid development and structured application.

Both application were developed with ASP.NET and C#.NET languages and assistance of the tools Microsoft Visual Studio and Microsoft SQL Server.

KEYWORDS:

CRM (*Customer Relationship Management*), Avaliação de Desempenho 360°, C#.NET, ASP.NET, *Web*

Índice

LISTA DE FIGURAS	X
LISTA DE TABELAS.....	XII
ACRÓNIMOS.....	XIV
CAPÍTULO 1. INTRODUÇÃO	1
1.1 MOTIVAÇÃO.....	2
1.1.1 SigAV360 (Sistema Integrado de Gestão - Avaliação de Desempenho 360°).....	2
1.1.2 SigCRM (Sistema Integrado de Gestão – Customer Relationship Management).....	2
1.2 OBJECTIVOS	3
1.2.1 SigAV360 (Sistema Integrado de Gestão - Avaliação de Desempenho 360°).....	3
1.2.2 SigCRM (Sistema Integrado de Gestão – Customer Relationship Management).....	4
1.3 ORGANIZAÇÃO DO DOCUMENTO	6
CAPÍTULO 2. OBJECTIVOS, METODOLOGIA E PLANEAMENTO	7
2.1 CONTEXTO SUBJACENTE E OBJECTIVOS	7
2.1.1 SigAV360 (Sistema Integrado de Gestão - Avaliação de Desempenho 360°).....	7
2.1.2 SigCRM (Sistema Integrado de Gestão – Customer Relationship Management).....	9
2.2 METODOLOGIA DE DESENVOLVIMENTO	11
2.3 PLANEAMENTO GERAL	12
2.3.1 Confrontação com o plano de trabalho inicial.....	13
CAPÍTULO 3. TRABALHO REALIZADO	14
3.1 FERRAMENTAS DE DESENVOLVIMENTO	14
3.1.1 SigAV360 (Sistema Integrado de Gestão - Avaliação de Desempenho 360°).....	14
3.1.2 SigCRM (Sistema Integrado de Gestão – Customer Relationship Management).....	15
3.2 FRAMEWORK’S INTERNAS	15
3.2.1 CG.Framework.Web.....	16
3.2.2 CG.Framework.....	16
3.2.3 DAL (Data Access Layer).....	17
3.3 TRABALHO REALIZADO	17
3.2.1 Fase 1 – Integração e ambientação na empresa	17
3.2.2 Fase 2 – SigAV360 (Sistema Integrado de Gestão - Avaliação de Desempenho 360°).....	18
3.2.2.1 Análise	19
3.2.2.2 Alteração e definição do modelo de dados e implementação das funcionalidades	20
3.2.2.3 Funcionalidades	21
3.2.2.3.1 Funcionalidades no Back-end	21
3.2.2.3.2 Funcionalidades no Front-end	28
3.2.2.3.3 Funcionalidades gerais.....	31

3.2.2.4	Testes	33
3.2.3	<i>Fase 3 – SigCRM (Sistema Integrado de Gestão – Customer Relationship Management)</i>	34
3.2.3.1	Funcionalidades	35
3.2.3.1.1	Funcionalidades gerais	35
3.2.3.1.2	Funcionalidades implementadas	40
3.2.3.1.2.1	Funcionalidades de conceitos de CRM	40
3.2.3.1.2.2	Outras funcionalidades	45
3.2.3.2	Testes	50
CONCLUSÃO		51
4.1	SUMÁRIO.....	51
4.2	COMENTÁRIO CRÍTICO	52
4.3	TRABALHO FUTURO	53
ÍNDICE REMISSIVO.....		57
BIBLIOGRAFIA		59

Lista de Figuras

FIGURA 1 – ESQUEMA DA METODOLOGIA SCRUM	12
FIGURA 2 – MENU INICIAL	23
FIGURA 3 – MENU DO PROCESSO DE AVALIAÇÃO	23
FIGURA 4 – MENU DO QUESTIONÁRIO	23
FIGURA 5 – PÁGINA INICIAL DO <i>FRONT-END</i>	29
FIGURA 6 – CONFIGURAÇÃO DE COOKIES NO FICHEIRO ". CONFIG"	36
FIGURA 7 – DIAGRAMA DE CLASSES BASE USADAS NAS PÁGINAS <i>WEB</i> DA APLICAÇÃO SIGCRM	39
FIGURA 8 – CONTROLO DE PESQUISA DE ENTIDADES.....	41
FIGURA 9 – <i>TABS</i> EXISTENTES NA OPORTUNIDADE.....	42

Lista de Tabelas

TABELA 1 – PLANO DE TRABALHO FINAL	13
--	----

Acrónimos

AJAX	Asynchronous JavaScript and XML
ASP	Active Server Pages
AV360	Avaliação de Desempenho 360°
CRM	Customer Relationship Management
DAL	Data Access Layer
DLL	Dynamic Linked Library
ERP	Enterprise Resource Planning
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IIS	Internet Information Services
LOV	List of Values
MEI	Mestrado em Engenharia Informática
PEI	Projecto em Engenharia Informática
SIG	Sistema Integrado de Gestão
SigAV360	Aplicação de Avaliação de Desempenho 360° do ERP Sig
SigCRM	Aplicação CRM do ERP Sig
SigGP	Aplicação de Gestão Pessoal do ERP Sig
SQL	Structured Query Language
URL	Uniform Resource Locator
VB	Visual Basic
VBA	Visual Basic for Applications
XML	Extensible Markup Language

Capítulo 1

Introdução

Este capítulo pretende apresentar sucintamente os dois projectos, SigAV360 (Sistema Integrado de Gestão – Avaliação de Desempenho 360°) e SigCRM (Sistema Integrado de Gestão – *Customer Relationship Management*), definir a motivação destes, os seus objectivos e um resumo do trabalho desenvolvido na instituição.

Este estágio, realizado numa empresa externa à Faculdade de Ciências, faz parte da disciplina de Projecto em Engenharia Informática (PEI) do Mestrado em Engenharia Informática (MEI) da Faculdade de Ciências da Universidade de Lisboa. O objectivo é a realização de um projecto de Engenharia Informática na instituição de acolhimento (Capgemini Portugal), com âmbito e complexidade adequada ao MEI, durante do 2º ano do curso, ou seja, 9 meses.

Conforme a especificação estipulada pela faculdade, este projecto consiste na integração na empresa de acolhimento, num aprofundamento dos conhecimentos técnicos/científicos, aperfeiçoamento na capacidade de tomada de decisões, realização de trabalhos na instituição, contacto com a documentação técnica, contacto com novas tecnologias e/ou ferramentas de trabalho, aprofundamento das capacidades de redacção de relatórios e apresentação pública dos resultados obtidos.

Os objectivos consistem em aprofundar os acontecimentos adquiridos ao longo da faculdade em dois projectos, um de melhoramento de um sistema de avaliações de desempenho de 360°, denominado SigAV360 (Sistema Integrado de Gestão - Avaliação de Desempenho 360°) e outro, integrado numa equipa de desenvolvimento, a criação de um novo módulo da instituição, o módulo de CRM (*Customer Relationship Management*), intitulado SigCRM (Sistema Integrado de Gestão - *Customer Relationship Management*).

1.1 Motivação

1.1.1 SigAV360 (Sistema Integrado de Gestão - Avaliação de Desempenho 360°)

Hoje em dia, os profissionais de recursos humanos efectuem com determinada regularidade avaliações de desempenho. Estas avaliações são realizadas entre os profissionais pertencentes ao mesmo círculo empresarial, isto é, empregados, colaboradores, parceiros, fornecedores, superiores, subordinados ligados à empresa. Assim, pode-se avaliar as competências de cada profissional, permitindo desta forma dar sugestões de desenvolvimento e aperfeiçoamento, para aumentar a eficácia e a qualidade de vida no trabalho.

Tradicionalmente os avaliadores/avaliados respondem às avaliações em folhas, o que causa um amontoado de pilhas de papel e obriga a um maior número de pessoas para transcrever os dados para folhas de cálculo, potenciando a margem de erro, podendo tornar o resultado final não fiável.

A unidade das Aplicações Standard da Capgemini Portugal, para responder às necessidades de um cliente, desenvolveu uma aplicação chamada SigAV360, que permite gerir as avaliações de desempenho das empresas. Com o tempo e a necessidade crescente das empresas em sistemas de avaliações de desempenho, a Capgemini chegou à conclusão que o sistema desenvolvido era demasiado específico, e que teria de se reformular a aplicação e acrescentar novas funcionalidades.

1.1.2 SigCRM (Sistema Integrado de Gestão – *Customer Relationship Management*)

O projecto SigCRM, módulo de CRM da Capgemini, surgiu devido à necessidade de as empresas quererem melhorar a relação com os clientes e gerir o dia-a-dia com o cliente.

Actualmente, as empresas reconhecem que para alcançar uma vantagem competitiva sustentável devem tornar-se empresas dirigidas ao cliente. Devem trabalhar e fazer todos os esforços para agradar ao cliente. Como estas empresas mantêm e crescem de um modo não sustentado a sua carteira de clientes e de participações, acabam por não conseguir manter os antigos clientes e obter novos. Tudo isto por não possuírem um bom sistema capaz de gerir a informação relativa aos seus clientes, os seus hábitos, preferências e o seu historial.

Como isto não acontece, os profissionais destas empresas ao efectuarem um contacto comercial com um determinado cliente desconhecem se este se trata de um cliente com assuntos pendentes com a empresa, ou se é um cliente que não contacta a empresa há vários meses. Obrigando assim a justificações intermináveis por parte do cliente, que apenas demonstra a pouca atenção e importância que lhe dão.

Por vezes, em grandes empresas, com clientes de longa data, que deveriam de ter um atendimento diferenciado, personalizado, acabam por não ter a devida atenção. O cliente sente-se insatisfeito com o serviço que lhe é prestado, podendo-se assim perder um bom cliente.

O SigCRM vem colmatar este risco, proporcionando às empresas uma gestão dos clientes, disponibilizando de forma centralizada e acessível a toda a organização a informação actual e essencial de cada cliente. Desta forma, é mais fácil obter dados actualizados sobre os clientes, futuras oportunidades e respectivo acompanhamento das mesmas.

Este controlo e disponibilidade de informação, em conjunto com a gestão adequada da força de vendas, vai permitir um melhor seguimento das oportunidades em curso, ajudar a fechar negócios, a controlar melhor as vendas e a um melhor aproveitamento da informação disponível no suporte às decisões de negócio.

1.2 Objectivos

1.2.1 SigAV360 (Sistema Integrado de Gestão - Avaliação de Desempenho 360°)

A aplicação SigAV360 foi desenvolvida para ir de encontro às necessidades de um cliente, o cliente X (será denominado cliente X, por motivos de confidencialidade).

O cliente X necessitava de um sistema que tornasse os processos de avaliação de desempenho mais fiáveis, menos morosos, que despendessem menos recursos para analisar os dados. Portanto contactou a unidade de Aplicações Standard da Capgemini e informou-lhes do seu interesse.

Assim surgiu a aplicação SigAV360, uma aplicação *Web* desenvolvida à medida para o cliente X. Uma aplicação que torna as avaliações de desempenho da empresa num processo dinâmico, pessoal e automatizado no que toca na geração de relatórios estatísticos sobre as avaliações.

Hoje em dia, a Capgemini reconhece que esta aplicação deveria ter sido uma aplicação *standard* e não específica para o cliente. Por isso surgiu a necessidade de reformular a toda a aplicação, dotando-a de novas funcionalidades, tornando-a mais fácil de compreender e de manusear.

Assim os objectivos deste projecto são tornar a aplicação *standard*, com um sistema de navegação autónomo, independente do browser, a par da inclusão de pontos de referência relativos às páginas de navegação, melhorar o aspecto visual da aplicação, incluir novas funcionalidades, tais como a importação de empregados do módulo SigGP (Sistema Integrado de Gestão - Gestão de Pessoal) da empresa, consultar os relatórios estatísticos das avaliações já concluídas, duplicar avaliações, entre outras menos relevantes.

Para tornar a aplicação *standard*, é necessário efectuar grandes alterações ao modelo de dados actual, alteração/criação de tabelas, *views* sobre tabelas e *procedures*. Todas estas alterações desencadeiam alterações do código existente, exigindo uma percepção e adaptação ao código já implementado.

1.2.2 SigCRM (Sistema Integrado de Gestão – *Customer Relationship Management*)

O módulo SigCRM da Capgemini, será uma aplicação desenvolvida em ambiente *Web*, o que a torna flexível e facilmente escalável para satisfazer as necessidades dos clientes mais exigentes.

Ao ser desenvolvido em ambiente *Web*, a actualização de novas versões faz-se apenas num único local (servidor), não havendo necessidade de percorrer os computadores de todos os utilizadores, um a um, para actualizar a nova versão. Outras das vantagens de ser *Web-based* passa por os utilizadores poderem aceder ao CRM a partir de qualquer local, inclusive das suas casas, garantindo a segurança a partir de canais seguros, como por exemplo por VPN (*Virtual Private Network*) segura¹.

¹ VPN (Virtual Private Network) segura – É uma rede de comunicação privada, normalmente utilizada por empresas e/ou instituições, construída em cima de uma rede de comunicações pública (Internet). Com protocolos de criptografia ou autenticação consegue-se uma VPN segura.

Desta forma os objectivos deste projecto são de criar um módulo CRM que permita a manutenção dos dados de uma forma centralizada acessível a todos os utilizadores da aplicação, permitindo a que qualquer organização consiga os seguintes pontos:

- Gestão de contactos e relacionamentos;
- Registo de actividades de toda a organização;
- Manter relatórios de visita aos clientes;
- Gestão de processos;
- Agenda / Planeamento;
- Gestão de força de vendas;
- Gestão de oportunidades de vendas;
- Gestão de objectivos de venda;
- Análise de performance;
- Análise de estatísticas;
- Criação de MailingList;
- Sincronização de tarefas com a aplicação Microsoft Outlook;
- Receber alertas via correio electrónico de tarefas pendentes, finalizadas ou alteradas, oportunidades conseguidas ou possíveis oportunidades, para além de um resumo destas com determinada frequência, para utilizadores específicos.

Os dois projectos são desenvolvidos em ambiente *Web* utilizando as tecnologias mais recentes da Microsoft, tais como ASP.NET 1.0 e 2.0 (o módulo de SigCRM será com a versão 2.0), a linguagem de programação C#.NET e apoio de JavaScript.

O primeiro projecto (SigAV360), devido às suas pequenas dimensões e pouca especificidade, requer apenas uma pessoa para o desenvolver, já o segundo projecto (SigCRM), devido a ser um projecto a desenvolver de raiz e de grandes dimensões é integrado numa equipa de cinco elementos, incluindo um chefe de equipa.

1.3 Organização do documento

Este documento está estruturado da seguinte forma:

- O capítulo 1 que representa um capítulo introdutório que pretende apresentar sucintamente os dois projectos, definir a motivação destes, os seus objectivos e um resumo do trabalho desenvolvido na instituição.
- No capítulo 2 são descritos em pormenor os objectivos, planeamento, metodologia do desenvolvimento e contexto dos dois projectos, o SigAV360 e o SigCRM. Também é feita uma confrontação com o plano de trabalho inicial analisando as razões de eventuais desvios ocorridos.
- O capítulo 3 descreve as ferramentas usadas e o trabalho realizado nos dois projectos inseridos no âmbito do PEI. Este capítulo apresenta concretamente o que foi efectuado nas duas aplicações e na aplicação SigCRM, que foi integrado numa equipa, é apresentado o trabalho efectuado por mim.
- O capítulo final será o capítulo 4 onde é apresentado um sumário do trabalho realizado no âmbito do PEI, um comentário crítico e ainda o trabalho futuro que se pode realizar nos projectos.
- No fim segue os acrónimos, índice remissivo e bibliografia.

Capítulo 2

Objectivos, Metodologia e Planeamento

Neste capítulo são descritos em pormenor os objectivos, planeamento, metodologia do desenvolvimento e contexto dos dois projectos, o SigAV360 e o SigCRM. Também é feita uma confrontação com o plano de trabalho inicial analisando as razões de eventuais desvios ocorridos.

2.1 Contexto subjacente e objectivos

2.1.1 SigAV360 (Sistema Integrado de Gestão - Avaliação de Desempenho 360°)

Hoje em dia, os departamentos de Recursos Humanos da grande parte das empresas, principalmente as médias e grandes empresas, necessitam de realizar com frequência avaliações de desempenho internas. Mas sem um sistema que consiga gerir todo o processo das avaliações, desde a criação dos questionários, a gerar as avaliações entre os colaboradores e finalmente a análise os resultados, os Recursos Humanos têm de despender a quase totalidade dos seus recursos apenas no processo de transcrição dos dados de papel para folhas de cálculo e a futura análise destes, potenciando assim a margem de erro, tornando o resultado final não fiável.

No fim do processo, os resultados das avaliações são enviados por correio electrónico ou carta para os respectivos avaliados/avaliadores do processo de avaliação. Cada colaborador tem acesso às avaliações de desempenho efectuadas pelos seus avaliadores e aos objectivos que deverá cumprir até à próxima avaliação de desempenho. Mas como os dados foram enviados por correio electrónico ou por carta, existe a possibilidade destes se perderem e o avaliado não ter novamente acesso às avaliações.

Já as chefias que tomam decisões de gestão pessoal terão de receber um relatório completo de cada empregado, o que faz com seja alocado mais pessoal para fazer este tipo de relatórios para as chefias.

O cliente X (será denominado cliente X, por motivos de confidencialidade) contactou a unidade das Aplicações Standard da Capgemini, de forma a encontrar um sistema que respondesse às suas necessidades.

Desenvolveu-se assim a primeira versão do SigAV360, uma aplicação à medida do cliente X em ambiente *Web*, sendo assim acessível em qualquer local. É uma aplicação que torna as avaliações de desempenho da empresa num processo dinâmico, pessoal e automatizado no que diz respeito à geração de relatórios estatísticos sobre as avaliações.

Actualmente, a Capgemini reconhece que esta aplicação deveria ter sido uma aplicação *standard* e não específica para o cliente. Por isso surgiu a necessidade de reformular a toda a aplicação, dotando-a de novas funcionalidades, tornando-a mais fácil de compreender e de manusear.

Elaborou-se um documento com os requisitos necessários para tornar a aplicação o mais *standard* possível. Foram efectuadas algumas reuniões, tanto com o cliente X, para reconhecer quais as melhorias necessárias, tanto com alguns dos colaboradores da Capgemini, para fornecerem informações sobre o processo de avaliação de desempenho de outros Recursos Humanos.

Um dos principais objectivos seria permitir visualizar os relatórios estatísticos das avaliações já concluídas, ou seja, permitir às chefias consultar o resumo estatístico das avaliações concluídas. Esta funcionalidade não ficou implementada na primeira versão da aplicação.

Um dos problemas encontrados na aplicação existente é o que se pode designar de “*Lost in the Hyperspace*”, termo usado em sistemas hipermédia, mas aplicável em aplicações *Web* quando os utilizadores seguem vários *links* nas páginas e acabam por perder a noção, a consciência de onde estão, de onde vieram, o que pretendiam fazer. Portanto, um dos objectivos é implementar o sistema de navegação da aplicação e colocar pontos de referência relativos às páginas de navegação.

Um processo moroso na aplicação era a inserção de todos os colaboradores (empregados), tendo que se inserir um a um e atribuir para cada um os acessos seguros à página pessoal destes. Passou-se a chamar à página pessoal de cada colaborador de *Front-end*, pois é a partir desta que se tem acesso à base das avaliações de desempenho, ou seja, à consulta/resposta de avaliações e à consulta dos relatórios estatísticos.

Como esta aplicação faz parte de um dos módulos do SIG (Sistema Integrado de Gestão), a nova versão da aplicação terá a funcionalidade de importar os empregados do módulo SigGP, evitando assim este processo moroso e permitindo aos utilizadores registarem-se na página de acesso ao *Font-end*.

Este é o objectivo principal: melhorar a actual aplicação, tanto a nível de desempenho, como a nível funcional, tornando assim o processo de gestão de avaliações ainda mais automatizado e menos moroso.

Todas as funcionalidades a implementar na nova versão da aplicação são:

- Sistema de navegação autónomo, independente do *browser*, a par da inclusão de pontos de referência relativos às páginas de navegação;
- Modificar o aspecto visual da aplicação;
- Duplicação de um processo de avaliação já existente;
- Permitir consultar os relatórios estatísticos das avaliações já concluídas;
- Pesos (percentagem) por tipo de relação;
- Definir um número pretendido de avaliações por avaliador/avaliado;
- Nova funcionalidade que informa das incoerências de avaliações geradas que existem num processo de avaliação;
- Criar empregados externos à empresa (exemplo: em consultoras, onde os empregados efectuem *Outsourcing* num cliente);
- Importação de empregados do módulo SigGP.

2.1.2 SigCRM (Sistema Integrado de Gestão – *Customer Relationship Management*)

Uma das grandes causas pela perda de clientes nas empresas, deve-se ao facto de a informação essencial ao cliente não estar acessível em qualquer altura/local de forma centralizada a toda a organização. O que acontece nestas empresas é que os profissionais, ao efectuarem um contacto comercial com um cliente, não têm conhecimentos suficientes sobre este. Pequenas informações como se é um cliente de longa data, os hábitos, as preferências, as oportunidades de negócio conseguidas/perdidas, os produtos adquiridos à empresa, todo o historial do cliente, são informações essenciais. O conhecimento do cliente evita obrigar o cliente a justificações intermináveis que demonstram a pouca atenção e importância que lhe dão, evita a perda de uma oportunidade de negócio, ou mesmo a possibilidade de conseguir uma nova venda.

O SigCRM proporciona às empresas uma gestão dos clientes apropriada, disponibilizando de forma centralizada e acessível a toda a organização a informação actual e essencial de cada cliente. Este sistema disponibiliza a informação necessária para conseguir chegar aos objectivos de venda do negócio, ganhar novas oportunidades e novos clientes.

Como as áreas de negócio dos clientes que possam vir a adquirir a aplicação são diferentes em muitos aspectos e, por isso, a informação que pretendem neste tipo de aplicações é variável, este sistema é desenvolvido com um conjunto de funcionalidades *standard*. Futuramente se o cliente pretender podem ser acrescentadas novas funcionalidades de forma a reunir a informação mais específica da sua área de negócio.

Sendo assim, as funcionalidades *standard* do SigCRM são:

- Gestão do relacionamento com os contactos de uma empresa, sejam eles clientes, fornecedores ou outros;
- Possibilidade de gestão da força de vendas através do registo de oportunidades identificadas e objectivos de venda;
- Gestão de actividades;
- Criação de relatórios de visita;
- Agenda dos utilizadores e sincronização do planeamento pessoal de cada um com a aplicação Microsoft Outlook;
- Efectuar análises de performance e estatísticas;
- Gestão de despesas da empresa;
- Registo de contactos recebidos/effectuados;
- *Mailing List*;
- Manutenção dos dados de acesso à aplicação, para cada utilizador (*password* e *username*);
- Gerir despesas internas;
- Gerir a facturação do negócio.

2.2 Metodologia de desenvolvimento

A metodologia de desenvolvimento usada foi a metodologia SCRUM, que se descreve de seguida.

SCRUM é uma metodologia de desenvolvimento que se baseia nos seguintes conceitos:

- **Proprietário do projecto** – Para cada projecto existe uma pessoa responsável, tipicamente denominada como o proprietário do projecto. As suas funções passam por atribuir prioridades às tarefas que compõem o projecto e definir quais as tarefas que devem ser incluídas num determinado ciclo de desenvolvimento do projecto.
- **Lista de tarefas** – A lista de tarefas que compõem o projecto é outro dos conceitos desta metodologia. Esta lista é organizada por prioridades, isto porque existe a necessidade de compreender quais as actividades que realmente são mais importantes para o desenvolvimento do projecto. Como a lista é criada a par da criação do primeiro planeamento do projecto, ela é dinâmica, porque é impossível ter uma noção total de todas as tarefas necessárias à conclusão do projecto numa fase inicial.
- **Sprint** – O ciclo de realização das tarefas denomina-se *sprint*. Este não é mais do que um período de tempo em que a equipa realiza as tarefas a que se propôs. Estas tarefas são definidas durante a reunião de planeamento. Durante a reunião, os membros da equipa e o proprietário do projecto chegam a um acordo sobre as tarefas a realizar até ao final do *sprint*. Os tempos de realização são unicamente estimados pela equipa, o proprietário do projecto terá de influenciar a equipa a realizar as tarefas que possam ser deixadas fora do *sprint* e as que são mais importantes, nem que seja necessário, por exemplo, reduzir o âmbito de uma funcionalidade de forma a diminuir a estimativa da sua conclusão e assim poder-se realizar outra tarefa.
- **Scrum** – São reuniões diárias que existem durante um ciclo de desenvolvimento. Nestas reuniões é realizado o ponto de situação de cada elemento da equipa, o que conseguiu terminar, a tarefa com a qual está a ter dificuldades, o trabalho realizado no dia anterior e as expectativas para o dia de trabalho. Nestas reuniões está presente uma pessoa responsável por moderar a

reunião, designado de *scrum master*, que para além da sua função de moderar a reunião tem a função de impedir a equipa a se comprometer a realizar mais tarefas do que aquelas que tem capacidade. Esta decisão deve ser tomada durante as reuniões de planeamento.

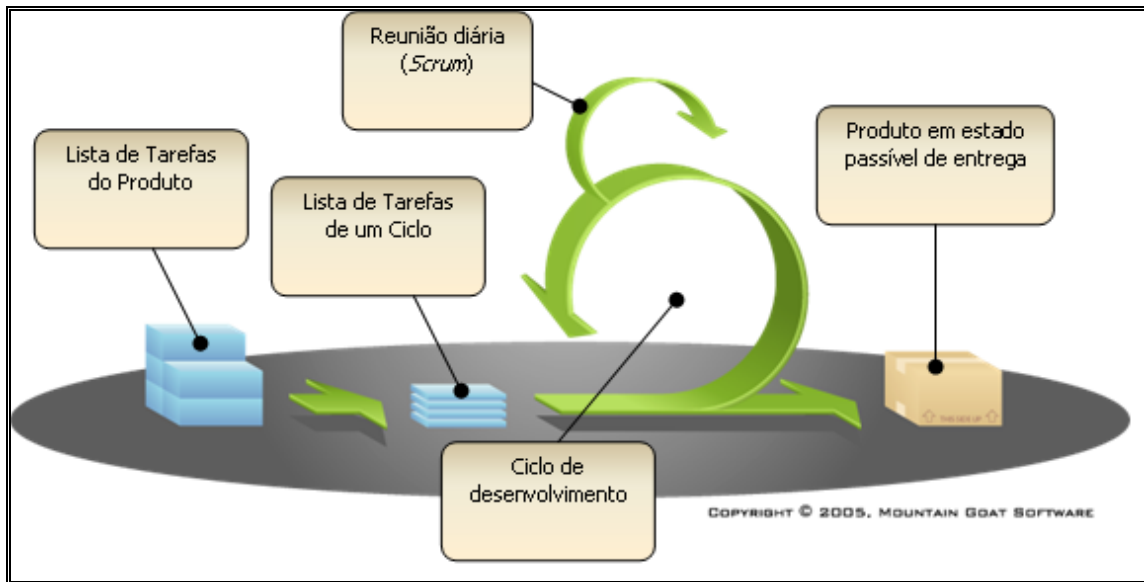


Figura 1 – Esquema da metodologia SCRUM

2.3 Planeamento geral

O plano de trabalho para o PEI dividiu-se em três fases separadas. A primeira fase não está ligada a nenhum dos projectos que constem neste relatório, é antes uma introdução à empresa, de ambientação e integração à empresa. A segunda fase abrange o trabalho de implementar novas funcionalidades na aplicação SigAV360. A terceira e última fase engloba o trabalho realizado no SigCRM.

A calendarização final destas fases está definida na tabela 1, que se segue:

Integração na instituição	<p>Formação – 2 semanas</p> <p>Implementação de um pequeno projecto – 3 semanas</p>
Desenvolvimento do primeiro projecto	<p>SigAV360 – 4 meses</p> <ul style="list-style-type: none"> • Análise • Alteração e definição do modelo de dados (Base de dados) • Implementação • Testes
Desenvolvimento do segundo projecto	<p>SigCRM – 4 meses (*)</p> <ul style="list-style-type: none"> • Implementação de funcionalidades • Testes

Tabela 1 – Plano de trabalho final

(*) A integração da aluna na implementação do projecto é efectuada aquando este já está em curso, desta forma não participa nas fases de análise e definição do modelo de dados (Base de Dados), não excluindo a sua participação em alterações ao modelo de dados.

2.3.1 Confrontação com o plano de trabalho inicial

Como planeamento de trabalho inicial, estava previsto como segundo projecto o desenvolvimento de um Portal do Empregado, mas à chegada da data de início deste, tal não se verificou. As razões que se podem enunciar são inúmeras, a principal prende-se com inexistência de um comprador para o projecto idealizado. Como tal não se verificou, optou-se pelo SigCRM, um projecto com aceitação pelos clientes do Departamento de Aplicações Standard da Capgemini.

Capítulo 3

Trabalho realizado

Este capítulo descreve as ferramentas usadas e o trabalho realizado nos dois projectos inseridos no âmbito do PEI.

3.1 Ferramentas de desenvolvimento

A principal ferramenta utilizada foi a aplicação Microsoft Visual Studio.NET, trabalhando-se sobre a Framework.NET, envolvendo as tecnologias C#.NET, ASP.NET e JavaScript.

Pertencente ao departamento da Unidade de Aplicações Standard da Capgemini, foram usadas algumas das *Framework's* desenvolvidas pela equipa .NET. Sendo elas: a CG.Framework.Web, um modelador da base de dados para cada aplicação chamado de DAL (*Data Access Layer*) e a CG.Framework que efectua a ligação com o DAL. Falo mais especificamente destas *Framework's* na secção 3.2 “*Framework's* internas”.

Foi usada uma ferramenta de controlo de versões, a aplicação Microsoft Source Safe que tem integração com a aplicação Microsoft Visual Studio.NET. De forma a utilizar as aplicações em ambiente *Web* foi utilizado o IIS (*Internet Information Services*), servidor *Web* para gerar as páginas HTML dinâmicas, como o ASP. Para o repositório de dados foi usado a aplicação Microsoft SQL Server.

Os dois projectos não foram desenvolvidos com as mesmas versões das ferramentas, as versões das ferramentas utilizadas foram:

3.1.1 SigAV360 (Sistema Integrado de Gestão - Avaliação de Desempenho 360°)

- Microsoft Visual Studio .NET 2003
- Framework .NET 1.1
- ASP.NET 1.1
- Microsoft Source Safe 6.0

- Pertencentes ao departamento da Unidade das Aplicações Standard da Capgemini: CG.Framework.UI, CG.Framework e SigAV360.DAL compatíveis com a Framework.NET 1.1

3.1.2 SigCRM (Sistema Integrado de Gestão – *Customer Relationship Management*)

- Microsoft Visual Studio .NET 2005
- Framework.NET 2.0
- ASP.NET 2.0
- Microsoft Source Safe 2005
- Pertencentes ao departamento da Unidade das Aplicações Standard da Capgemini: CG.Framework.Web, CG.Framework e SigCRM.DAL compatíveis com a Framework .NET 2.0

Apesar dos dois projectos usarem versões diferentes de Microsoft SQL Server, uma das preocupações foi o uso da mesma sintaxe SQL para que as aplicações funcionassem em ambas as versões de Microsoft SQL Server.

Fora estas aplicações e na fase inicial do PEI, aquando da fase de integração na empresa, para elaboração de um pequeno projecto, foram usadas as ferramentas Microsoft SQL Server 2003, a aplicação SIG da Capgemini e a linguagem de programação VBA (*Visual Basic for Applications*).

3.2 *Framework's* internas

Nesta secção irei descrever as *Framework's*, pertencentes à Unidade das Aplicações Standard da Capgemini, que forma utilizadas nas duas aplicações SigAV360 e SigCRM. Estas duas aplicações usam as mesmas *Framework's* excepto na CG.Framework.Web, pois existem duas versões desta *Framework*, versão em .NET 1.1 e em .NET 2.0. Como o SigAV360 foi desenvolvido com a Framework.NET 1.1 teve de se usar a CG.Framework.Web em 1.1 devido às compatibilidades.

3.2.1 CG.Framework.Web

Apesar de existirem duas versões desta *Framework*, estas representam o mesmo, ou seja, representam todos os conceitos UI (*User Interface*).

As grelhas, os botões, os *links*, as *combo-box*, *list-box*, todas as classes que representam as páginas base de uma aplicação (existem as páginas de listagem, *ListPageBase* e as páginas que representam um registo de uma listagem, *PageBase*, ver secção 3.2.3.1.1 “Funcionalidade Gerais”), as caixas de texto como os vários tipos (numérica, decimal, percentagem, texto), os controlos do tipo data, entre outros, fazem parte desta *Framework*.

Com esta *Framework* a estrutura base da cada página já está definida, tal como os controlos das páginas, caso se pretenda mais funcionalidades ou modificar alguma propriedade ou estrutura, basta usar o conceito de herança e criar controlos ou páginas específicas para a própria aplicação.

3.2.2 CG.Framework

Esta *Framework* contém classes de controlo, que podem ser úteis nas aplicações .NET, independentemente do tipo de aplicação (*Web* ou *Windows Forms*).

Por exemplo, existe uma camada que controla os recursos das aplicações. Os recursos são utilizados para simplificar e automatizar o processo de traduzir toda a aplicação para a linguagem usada (português, inglês,...). Os recursos são inseridos na base de dados da aplicação e para cada chave é associado o seu valor e linguagem (português, inglês,...). Os controlos que fazem parte da CG.Framework.Web ou que herdem destes, têm uma propriedade designada de “TextKey” ao qual é associado a chave do recurso. A camada controladora dos recursos, identifica a linguagem em que a aplicação se encontra e associa o valor correcto para a chave.

No caso da aplicação SigAV360 esta camada de controlo dos recursos ainda não existia, foi criada para a aplicação SigCRM, por isso o SigAV360 utiliza um sistema diferente para controlar os recursos, baseia-se em ficheiros de recursos (ver secção 3.2.3.1.1 “Funcionalidades gerais”).

Esta *Framework* também tem as classes necessárias para efectuar ligações a uma base de dado com base numa conexão, classes que permitem executar *Stored Procedures* e

classes que conjuntamente com instâncias a classes do DAL permite inserir, apagar ou modificar dados de uma base de dados.

3.2.3 DAL (*Data Access Layer*)

O DAL representa a camada de acesso a dados, que permite a modelação da base de dados por objectos.

Todas as classes desta *Framework* são geradas por uma aplicação interna do departamento, que com base numa conexão a uma base de dados, gera para cada tabela quatro classes que permitem modificar os dados dela.

Como o DAL é gerado para cada base de dados cada aplicação tem o seu próprio DAL. O DAL do SigCRM designa-se SigCRM.DAL.

Tal como já referi esta *Framework* funciona conjuntamente com a CG.Framework, que entre outras classes contem as necessárias para inserir, apagar ou modificar os dados de uma base de dados. Por exemplo, para se inserir uma entidade numa tabela da aplicação SigCRM teria de se criar uma instância da classe do DAL que representa uma entidade, associar os dados da entidade e por fim, devido à herança existentes nas classes, pode-se invocar o método *Insert()*, pertencente à uma classe da CG.Framework chamada *Entity*, que insere os dados da entidade na tabela.

3.3 Trabalho Realizado

Irei dividir esta secção por três fases, que correspondem às fases mencionadas no capítulo 2 no planeamento geral (secção 2.3), sendo assim mais fácil seguir o trabalho que foi realizado na Capgemini no âmbito do PEI.

3.2.1 Fase 1 – Integração e ambientação na empresa

Na fase inicial do PEI, na fase de integração e ambientação na empresa, com o objectivo de me familiarizar com algumas das ferramentas e linguagens de programação usadas na empresa, durante duas semanas estive abrangida numa formação pessoal, onde estudei as linguagens de programação C#.NET, Visual Basic .NET e ASP.NET, bem como a ferramenta Microsoft Visual Studio .NET 2003.

Como apoio para a minha formação individual usei o curso do *MSDN training*, “*Developing Microsoft ASP.Net Web Applications using Visual Studio .Net*”,

juntamente com o livro que acompanha o curso. Este curso ajudou-me imenso a aprofundar os conhecimentos da ferramenta e linguagem de programação C# (tive algum contacto com esta linguagem de programação numa disciplina opcional do meu curso de Engenharia Informática na faculdade) e também a conhecer as restantes linguagens de programação (ASP.NET e VB.NET).

Ainda nesta primeira fase, de forma a conhecerem as minhas capacidades de ambientação e os meus conhecimentos, a equipa de desenvolvimento da Capgemini pediu-me para desenvolver um pequeno projecto. Este projecto seria desenvolvido em VBA, integrado com a aplicação Microsoft Excel e Microsoft Word. O projecto consistia em desenvolver um “mapa personalizado”², que, com base em filtros escolhidos pelo utilizador (data de início, data fim, facturas, cliente, firma, etc..), devolve uma lista de facturas ainda não pagas pelos clientes numa firma num período de facturação, com possibilidade de gerar as cartas de facturação em atraso ou enviar por correio electrónico um HTML da carta aos clientes. Esta aplicação permitia ao utilizador escolher um *template* de um documento em Microsoft Word, sobre o qual as cartas são geradas. Este pequeno projecto teve uma duração de 3 semanas, tal como fora planeado e foi integrado na aplicação SIG para uso interno do departamento.

O desenvolvimento deste “mapa personalizado”, permitiu-me conhecer a aplicação SIG da Capgemini, e familiarizar-me com a linguagem VBA e as suas funcionalidades com as ferramentas Microsoft Office. Como este pequeno projecto não faz parte da proposta com o PEI, não vou detalhar mais sobre ele.

3.2.2 Fase 2 – SigAV360 (Sistema Integrado de Gestão - Avaliação de Desempenho 360°)

Tal como planeado, o primeiro projecto no âmbito do PEI foi a implementação de novas funcionalidades na aplicação SigAV360.

Como já foi referido, a aplicação já existia, mas tinha sido feito à medida para um cliente, o cliente X. Por ser uma aplicação específica para este cliente, inicialmente o

² Mapa personalizado é o termo designado pela Capgemini para indicar os projectos desenvolvidos em VBA (Visual Basic for Applications), normalmente usam o Microsoft Excel ou Microsoft Word, depois de desenvolvidos são inseridos nas aplicações SIG (Sistema Integrado de Gestão)

seu nome não foi SigAV360, apenas AV360 (Avaliação de Desempenho 360°), mas com a necessidade de tornar a aplicação *standard* passou a intitular-se SigAV360.

Quero deixar claro que apesar deste projecto ser um projecto individual, nunca estive sozinha nas decisões referentes ao projecto. A equipa de desenvolvimento é constituída por várias equipas, para diferentes projectos, áreas e linguagens de programação. Como a linguagem de programação C#.NET e ASP.NET é algo recente na unidade das Aplicações Standard da Capgemini, a equipa que desenvolve C#.NET e ASP.NET, na altura em que fui integrada na empresa, era apenas constituída por um elemento, sendo essa pessoa o meu chefe de equipa, Daniel Vinagre e eu seria o segundo elemento da equipa.

O Daniel Vinagre acompanhou o meu estágio e geriu os dois projectos .NET que realizei, tomando as decisões principais sobre o planeamento, estruturas gerais da aplicação e modelo de dados.

3.2.2.1 Análise

Para tornar a aplicação *standard* teve de se proceder a uma análise do que estava implementado na aplicação, baseando-se nas funcionalidades da aplicação e no modelo de dados actual. Depois desta análise foi efectuado um estudo, podendo-se chamar um estudo de mercado.

Este estudo consistiu em algumas reuniões, tanto com o cliente X, para reconhecer quais as melhorias necessárias, tanto com alguns dos colaboradores da Capgemini, para fornecerem informações sobre o processo de avaliação de desempenho de outros recursos humanos. Nestas reuniões, estavam presentes por parte da unidade das Aplicações Standard da Capgemini, o meu chefe de equipa e eu. Estas reuniões permitiram elaborar um documento com os requisitos necessários para tornar a aplicação *standard*. Com este documento pode-se efectuar o passo seguinte, a alteração e definição do modelo de dados actual.

A duração estimada para esta análise foi de uma semana.

3.2.2.2 Alteração e definição do modelo de dados e implementação das funcionalidades

A alteração e definição do modelo de dados seria o passo seguinte à análise das funcionalidades, mas esta fase acabou por se realizar a par da fase de implementação das novas funcionalidades, pelo facto de que como a aplicação já estava implementada não foi necessário desenhar o modelo de dados de raiz, apenas redesenhar o modelo de dados existente, como adicionar tabelas, novos campos às tabelas já existentes e adicionar novas ligações entre tabelas ou alterar as existentes. O redesenhar do modelo de dados é algo que se faz com mais certezas à medida que se vai implementado novas funcionalidades, evitando assim alterar inúmeras vezes o modelo de dados por ser ter efectuado uma primeira análise errada.

Para poder explicar o meu trabalho realizado neste projecto e como a aplicação já existia, terei de explicar as funcionalidades de toda a aplicação e nas funcionalidades que implementei explicarei o modo como foi desenvolvido, para melhor se entender o projecto.

Toda a aplicação é acedida via *Web*, a aplicação é instalada num servidor da empresa, a aplicação fica alojada num directório virtual criado com o IIS (*Internet Information Services*) e é criada uma base de dados com os dados necessários para a aplicação funcionar (tabelas, *views*, *stored procedures*). Todos os utilizadores ligados à rede da empresa e que tenham acesso à máquina podem aceder à aplicação por um browser *Web*. A aplicação foi desenvolvida com o cuidado de poder ser visualizada em qualquer browser *Web*. A garantia que damos é que funciona com o Internet Explorer e o Mozilla Firefox, mas verificou-se em outros browsers e aparentemente não existem problemas.

O acesso à aplicação é feito por duas formas, o *back-end* que é a parte da aplicação direccionada aos administradores do sistema, às pessoas responsáveis por criar as avaliações de desempenho (os questionários, as avaliações existentes entre os empregados, etc...). Mais à frente explicarei melhor o processo todo de criar uma avaliação de desempenho no *back-end*. O outro acesso à aplicação é pelo *front-end*, este é o acesso que os empregados utilizarão para poder responder às avaliações que lhe estão atribuídas, consultar avaliações já concluídas, dar aprovação de avaliações e, se existirem permissões para tal, visualizar os relatórios estatísticos das avaliações de desempenho concluídas.

- **Cookies e variáveis de sessão (Sessions)** – Sempre que são efectuados *logins* em qualquer um dos modos de funcionamento da aplicação, são criados *cookies* associados aos utilizadores e variáveis de sessão, a informação neles contida é cifrada por razões de segurança, desta forma cada sessão tem um *time-out* por defeito de 20 minutos, em que caso não se registe nenhuma acção de resposta de uma página a sessão expira, obrigando a um novo *login*. Este sistema já estava implementado.

Um *cookie* é um par do tipo nome/valor armazenado na máquina cliente. Esta técnica é utilizada por defeito pelo ASP.NET para manter o estado de uma sessão. Os *cookies* podem ser utilizados para armazenar, temporariamente ou persistentemente, informações sobre o utilizador, como por exemplo os seus dados. Os temporários perdem-se quando é terminada uma sessão e os persistentes ficam armazenados fisicamente no disco do cliente.

Os *cookies* são armazenados numa colecção *HttpCookieCollection*. Os objectos *HttpRequest* e *HttpResponse* contêm uma colecção de *cookies*. O *HttpRequest* indica os que são enviados do cliente para o servidor, o *HttpResponse* representa os que são enviados do servidor para o browser.

3.2.2.3 Funcionalidades

3.2.2.3.1 Funcionalidades no *Back-end*

Para poder explicar melhor a aplicação vou começar por explicar como foi implementado o sistema de navegação autónomo, independente do browser, a par da inclusão de pontos de referência relativos às páginas de navegação, implementado para a nova versão da aplicação.

Na aplicação existem três sub-menus, horizontais:

- **Menu inicial** – onde se pode criar empregados e os processos de avaliação que existirão na empresa.
- **Menu do Processo de Avaliação** – dentro de um processo de avaliação (edição do processo de avaliação) o menu inicial é substituído por este menu que permite criar os tipos de relação, perfis de avaliação, relações de empregados,

questionários, estatísticas administrativas e associar empregados a perfis de avaliação para o processo de avaliação actual.

- **Menu do Questionário** – dentro de um processo de avaliação (edição do processo de avaliação) num dos questionários existente para o processo de avaliação (edição do questionário) o menu do processo de avaliação é substituído por este menu que permite criar as questões do questionário, criar avaliações entre empregados e visualizar as inconformidades que possam existir para o processo de avaliação, devido a não existirem as avaliações mínimas pretendidas para cada perfil de avaliação.

Inicialmente estes menus não eram substituídos e surgiam três menus, o que confundia um pouco o utilizador, é que este tinha de ter a noção que ao criar um questionário deveria associá-lo a um processo de avaliação. E que ao criar uma questão deveria de associá-la a um questionário. Desta forma surgiam mais erros de inserção, associando por exemplo as questões a questionários errados.

Por isso em vez de surgirem três menus ao mesmo tempo, surge apenas um, que será substituído pelo seguinte conforme se está dentro de um processo de avaliação ou dentro de um questionário.

Mas desta forma para o utilizador voltar ao menu inicial teria de usar a barra de navegação do *browser*, por vezes teria de voltar várias vezes para trás até chegar ao menu inicial. O mesmo aconteceria se quisesse voltar ao menu do processo de avaliação actual caso estivesse na edição de um questionário.

Por isso decidiu-se colocar uns pontos de referência nas páginas que permitem voltar para trás. Para isso implementou-se um menu base, horizontal, para todas as páginas que permite efectuar o *logout* e à medida que se navegava na página e dependendo da página em que se encontra o utilizador, é acrescentado no fim do menu um *link* para o menu inicial, outro para o processo de avaliação actual, e outro para o questionário actual, tendo como pontos de referência o início (*home*), o processo de avaliação e o questionário de avaliação.

Este menu, para além de permitir o utilizador navegar, dá-lhe uma percepção do processo de avaliação em que se encontra, ou do questionário, porque à frente do ponto de referência surge o nome do processo de avaliação ou do questionário.

As figuras seguintes ilustram os três casos possíveis dos pontos de referência:



Figura 2 – Menu inicial

Na figura 3, se se “clicar” no *link* do **Home** ou em **Processo de Avaliação de Desempenho** é se redireccionado para o menu que é ilustrado na figura 2, o *link* do **Processo de Avaliação de Desempenho** existe para se dar a percepção do processo de avaliação em que se encontra e para além desta percepção no menu inferior o rectângulo mais claro (azul claro) indica em que página o utilizador se encontra.

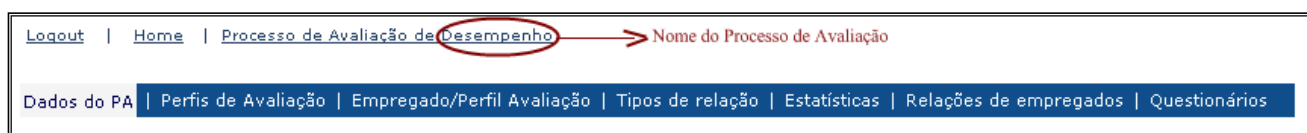


Figura 3 – Menu do processo de avaliação

Na figura 4, se se “clicar” no *link* do **Questionário de Desempenho** é se redireccionado para o menu ilustrado na figura 3 mas na página dos questionários (último *link* do menu ilustrado na figura 3).

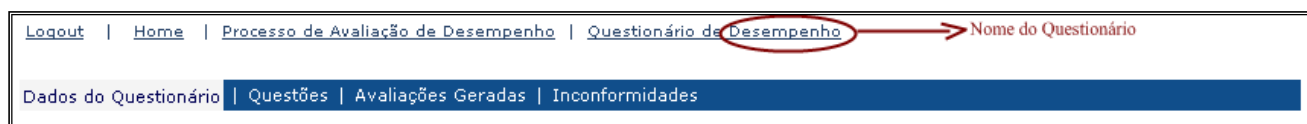


Figura 4 – Menu do questionário

Com isto pode-se resumir dizendo que a funcionalidade do sistema de navegação autónomo, a par da inclusão de pontos de referência relativos às páginas de navegação usa dois conceitos de sistemas hipermédia, o sistema de navegação e o sistema de orientação³.

De seguida descrevo resumidamente as funcionalidades existentes na aplicação, no *back-end*. Quando estas foram implementas por mim, descrevo-as em pormenor.

- **Login** – O *login* é único para os administradores. Uma vez efectuado o *login* válido é iniciada uma nova sessão. Com um *time-out* de sessão de 20 minutos, configurável.

³ H. Van Dyke Parunak – Hypermedia topologies and user navigation – Hypertext’89

Depois do *login* tem-se acesso ao menu inicial onde se pode inserir empregados e criar processos de avaliação.

- **Inserir empregados internos/externos** – O primeiro passo será inserir os empregados envolvidos em qualquer uma das avaliações. É mostrada uma listagem dos empregados que são internos à empresa e os externos (ex: colaboradores da empresa em *Outsourcing* num cliente).

A criação de um empregado passa pela inserção dos dados pessoais do empregado (nome, números de empregado, contribuinte, bilhete de identidade, área pertencente e cargo ocupado), criação de *password* de acesso ao *front-end* e também se o empregado será um avaliador e/ou avaliado nos processos de avaliação.

A funcionalidade de inserir empregados já existia, mas não era possível fazer uma distinção de empregado interno ou externo. Para isso à tabela que possui os dados dos empregados foi adicionado um campo que corresponde ao número de empregado.

Ainda nesta página de inserção de empregados passou a existir a nova funcionalidade de importação dos empregados da aplicação SigGP. O mecanismo, com base numa tabela de configuração existente no repositório de dados, executa um comando SQL com determinados filtros (basicamente um SELECT ao repositório de dados onde se encontra a aplicação do SigGP), para receber uma lista de empregados existentes no SigGP. Esta lista é disponibilizada ao utilizador permitindo-lhe escolher os empregados que quer importar, os que já foram importados não surgem nesta lista.

- **Criar um processo de avaliação** – É no processo de avaliação que se manuseiam todos os dados relativos a uma avaliação de desempenho, ou seja, é onde são indicadas as relações entre os empregados, os tipos de relações existentes, são criados os questionários para a avaliação e designadas as avaliações entre os empregados.

Nesta funcionalidade apenas se inserem dados indicativos do processo de avaliação, sendo esta uma funcionalidade que já existia. Foi acrescentada a esta funcionalidade, a opção de duplicar o processo de avaliação já existente, que cria um novo processo de avaliação com os mesmos empregados associados, as

mesmas relações entre eles, os mesmos tipos de relações, perfis de avaliação e questionários, tendo opção de duplicar também as avaliações existentes.

Ao se “clique” para editar um processo de avaliação, tem-se acesso ao menu do processo de avaliação onde se criam tipos de relação, perfis de avaliação, relações de empregados, estatísticas administrativas e associa-se empregados aos perfis de avaliação existentes no processo de avaliação.

- **Criar tipos de relação** – Permite criar os tipos de relação existentes numa empresa (por exemplo, Superior Hierárquico – Subordinado).

Foi acrescentada a esta funcionalidade a associação de um peso por tipo de relação, para fins de cálculos de percentagens nos relatórios estatísticos. Por exemplo, para uma dada pergunta de um questionário, se um superior hierárquico tem um peso de 70%, a sua resposta nessa pergunta terá mais valor. O total das percentagens de todos os tipos de relação deve de perfazer 100%, como é evidente.

Nesta funcionalidade, ao ser criado um tipo de relação também se pode dizer que esta relação tem uma relação implícita. Ou seja, pode-se criar um tipo de relação “Superior Hierárquico” e indicar que este tem uma relação implícita com o tipo de relação “Subordinado”.

- **Criar perfis de avaliação** – Permite criar os perfis de avaliação existentes para este processo de avaliação. Esta funcionalidade já existia.
- **Associar empregados a perfis de avaliação** – Esta funcionalidade já existia e permite associar os empregados a um perfil de avaliação existente.
- **Relações entre empregados** – Aqui indica-se quais as relações existentes entre os empregados. Para um dado empregado surge uma lista com os tipos de relação existentes, depois para cada tipo de relação associa-se um ou mais empregados, caso esta relação tenha uma relação implícita com outro tipo de relação, é automaticamente atribuído para a relação implícita o outro empregado. Um exemplo seria:

- O André é superior hierárquico do Bruno, ou seja o Bruno é subordinado do André. Existem os tipos de relação superior hierárquico e subordinado. Nas relações de empregados, no empregado André e no tipo de relação superior

hierárquico associa-se o Bruno. Depois de gravada esta relação, nas relações existentes no empregado Bruno, pode-se verifica-se que já existe o André no tipo de relação subordinado.

A funcionalidade de criar relações de empregados já existia, o mecanismo automático das relações implícitas é que não estava implementado.

- **Estatísticas** – Aqui podem-se visualizar/criar estatísticas administrativas. Estas estatísticas baseiam-se na execução de um *Stored Procedure* em SQL.

Como o cliente que adquire a aplicação não conhece o modelo de dados, não pode criar nenhuma estatística. Apenas a unidade das aplicações Standard da Capgemini o poderá fazer e com meios de *scripts* inserir no repositório de dados do cliente. O cliente depois só tem de criar uma estatística, dando-lhe um nome e dizendo o nome do *script* (*Stored Procedure* que se criou) a executar. Desta forma podem-se visualizar estatísticas administrativas, como por exemplo, o número de avaliações que ainda não foram respondidas, uma percentagem dos tipos de relação que responderam primeiro aos questionários, etc. Esta funcionalidade já estava implementada.

- **Indicar o número pretendido e mínimo de avaliações** – Esta funcionalidade já existia e permite para um perfil de avaliação, tipo de relação e questionário, indicar um número mínimo e pretendido de avaliações entre empregados. Este número deverá existir, caso não aconteça isto surgirá na listagem de inconformidades, referida mais à frente.

O número pretendido e mínimo de avaliações servem para a funcionalidade de gerar automaticamente as avaliações entre os empregados.

- **Criar questionários** – Ao ser criado um questionário, indica-se se este questionário terá relatórios estatísticos. Por exemplo num questionário de objectivos para o ano seguinte, não faz muito sentido visualizar-se percentagens e gráficos das questões por cada tipo de resposta/tipo de relação.

Também é indicado se o questionário necessita do conhecimento da pessoa avaliada. Num questionário de objectivos, ou mesmo de avaliação de desempenho pode fazer sentido existir a indicação de que a pessoa avaliada teve acesso aos seus objectivos/avaliação e que comentário fez à sua avaliação.

Esta funcionalidade não estava implementada, na parte do *back-end* apenas teve de se implementar duas *flags* no momento da criação de um questionário, na parte do *front-end* já se teve de implementar a aceitação do avaliado e o mecanismo de gerar relatórios estatísticos.

Depois da criação do questionário podem-se gerar as avaliações para o processo de avaliação e questionário.

Ao editar um questionário passa-se a ter acessível o menu do questionário, onde se podem criar as questões do questionário, gerar/criar as avaliações que existirão e visualizar as inconformidades existentes nas avaliações.

- **Questões** – A construção desta página usa um gerador⁴. A página inicialmente apresenta uma listagem dos tipos de questão disponíveis. Depois de um tipo de questão ser escolhido e a questão ser gravada, o gerador constrói a página *Web* que permite ao utilizador inserir as respostas possíveis para essa questão e caso se trate de uma questão do tipo tabela, permite também adicionar as sub questões. Existem tipos de questão de texto livre, de resposta longa e um tipo de questão que apenas permite inserir separadores num questionário (por exemplo, tópicos num questionário).

A funcionalidade de inserir questões de vários tipos já existia, mas foram necessários aperfeiçoamentos no gerador desta página pois existiam algumas incoerências em alguns tipos de questões. Principalmente nos tipos de questão que já têm respostas possíveis por defeito, por exemplo uma pergunta do tipo “Concordo/Discreto” já tem por *default* as respostas “Concordo totalmente”, “Concordo”, “Não concordo nem discreto”, “Discreto”, “Discreto totalmente”. Estas respostas eram inseridas na tabela no momento de instalação, mas devido às incoerências que surgiam, houve a necessidade de solucionar o problema criando duas tabelas, uma com o tipo de questão *default* outra com as respostas para cada tipo de questão *default*.

⁴ Gerador é designação dada ao código que é criado com base em classes. A parte de design da página (extensão “.aspx”) não contém quase nenhum controlo, a página é quase toda desenhada com base numa classe. Neste caso a classe, conforme o tipo de questão, carrega na página a estrutura da questão, e é sobre esta estrutura que as sub questões e respostas possíveis são adicionadas.

A outra funcionalidade necessária foi a possibilidade de dar uma pontuação às respostas, para poder realizar alguns dos cálculos estatísticos.

- **Avaliações geradas** – Esta funcionalidade já existia, permite visualizar as avaliações geradas para o questionário em questão e o seu processo de avaliação, permite também inserir novas avaliações ou apagar existentes.
- **Inconformidade** – Esta é uma nova funcionalidade que informa das incoerências de avaliações geradas que existem num processo de avaliação para o questionário actual. Se para um dado perfil de avaliação – tipo de relação associado a este questionário se estabeleceu um número mínimo de avaliações ao qual não se chegou, é mostrada a incoerência com um sinal de semáforo vermelho, caso se tenha conseguido chegar ao número mínimo. Mas não se conseguiu chegar ao número pretendido então é mostrada a incoerência com a cor amarela.

Com esta funcionalidade o utilizador pode criar corrigir as incoerências criando por exemplo novas avaliações.

3.2.2.3.2 Funcionalidades no *Front-end*

O *front-end*, como já foi referido, é o local onde os empregados poderão responder às avaliações, consultar avaliações já concluídas, dar aprovação de avaliações, e se tiverem permissões para tal, poderão visualizar os relatórios estatísticos das avaliações de desempenho concluídas.

- **Login** – O *login* é único para cada empregado. Uma vez efectuado o *login* válido é iniciada uma nova sessão. Ela tem um *time-out* de sessão configurável de 20 minutos.

O empregado em qualquer altura pode registar-se, basta que este esteja inserido na aplicação como empregado (a inserção é feita no *back-end*, na funcionalidade de inserir empregados ou na importação de empregados do módulo SigGP). O empregado também pode alterar/recuperar a *password*.

Esta funcionalidade já estava implementada, foi necessário fazer uma pequena alteração no *login*, porque o número de empregado não existia. Tomou-se esta decisão porque é mais fácil para o utilizador efectuar fixar o *login* com número de empregado.

Uma vez que seja efectuado o *login* com sucesso todo o processo de avaliação é iniciado, o empregado tem acesso às avaliações que deve de fazer, às avaliações que necessitam da sua aceitação (por exemplo, o empregado tem o direito de tomar conhecimento dos objectivos que lhe foram propostos) e de visualizar as avaliações que já foram efectuadas.

Para os empregados com acessos para tal, é permitido consultar os relatórios estatísticos dos processos de avaliação já concluídos.

Na figura 5 pode-se visualizar uma representação sem dados de um *front-end*, onde o empregado terá acesso às funcionalidades já mencionadas.

Escolhendo o processo de avaliação têm-se acesso às funcionalidades possíveis para o processo de avaliação. As funcionalidades surgem sobre a forma de grupos e em cada grupo é mostrada a lista de avaliações possíveis para o utilizador.

Processo de avaliação: Desempenho Ano: 2007

Formulários que Necessitam Aceitação

Avaliador	Formulário
Sem registos	Zona onde surge a listagem de avaliações que necessitam de aceitação

Formulários a Preencher

Nome do avaliado	Formulário
Sem registos	Zona onde surge a listagem de avaliações para responder

Relatórios de Avaliações

Nome do avaliado	Formulário
Sem registos	Zona onde surge a listagem dos relatórios estatísticos

Figura 5 – Página inicial do *front-end*

De seguida descrevo as funcionalidades disponibilizadas para o utilizador com uma sessão iniciada e com um processo de avaliação seleccionado.

- **Aceitar avaliações** – No primeiro grupo surge a listagem das avaliações para as quais o utilizador deve de dar a sua aceitação. Tal como já foi referido, pode-se criar um questionário que necessite da aceitação do avaliado (por exemplo os questionários de desempenho do ciclo de avaliação anterior ou o de objectivos para o próximo ciclo de avaliação). Nestes casos, depois de o avaliador submeter o questionário, o avaliado têm de dar a sua aceitação para finalizar a avaliação.

O avaliado indica se aceita ou não a avaliação efectuada e tem a possibilidade de acrescentar comentários.

Esta opção de aceitação, não rejeita a avaliação caso o avaliado não a aceite, apenas serve para fins estatísticos. Uma vez efectuada a aceitação não é possível alterá-la, apenas visualizá-la. Esta funcionalidade não existia na versão anterior e foi de fácil implementação.

- **Responder a avaliações (questionários)** – No segundo grupo surge a listagem que permite ao utilizador responder às avaliações.

Esta funcionalidade já existia e permite ao avaliador (utilizador com sessão iniciada) responder aos questionários. Dependendo do tipo de questionário/avaliação, o utilizador pode ter de definir os seus objectivos ou os do(s) avaliado(s) atribuído(s) a si, pode ter de efectuar uma auto-avaliação ou avaliar o seu(s) avaliado(s) ou por exemplo, simplesmente responder a um questionário de opiniões sobre a empresa.

- **Relatórios das avaliações** – Esta funcionalidade já existia, mas os relatórios estatísticos eram muito simples e apenas acessíveis ao utilizador administrador. Os relatórios estatísticos eram estatísticas simples para cada processo de avaliação. Por exemplo para um processo de avaliação e numa avaliação era apresentado o número de avaliações respondidas para cada questão da avaliação ou as respostas mais escolhidas. Não existia uma distinção das respostas dadas por tipo de relação dos avaliadores.

Criaram-se as restantes estatísticas por cada questão do questionário numa dada avaliação, sendo estas:

- Médias de desempenho por tipo de avaliador: Para uma questão é calculada para cada resposta a média de respostas dadas por tipo de relação e peso das respostas;
- Médias por tipo de relação: Para uma questão é calculada a percentagem de respostas por tipo de relação;
- Respostas dadas por tipo de relação: Nas questões que tenham texto livre, são mostradas para cada tipo de relação as respostas dadas;

- Médias por resposta: Nas questões do tipo tabela (várias perguntas para a questão, em que para cada uma destas têm várias respostas possíveis) é apresentada uma média de número de respostas dadas/percentagem para cada uma das perguntas. Esta estatística já existia;
- Número de avaliadores/tipo de relação para esta avaliação.

Dependendo do tipo de questão é apresentado uma, algumas ou todas estas estatísticas. Por exemplo para as questões do tipo texto livre apenas têm as respostas dadas por tipo de relação. Já para as questões que não sejam do tipo tabela, mas que contenham texto livre, são apresentadas todas as estatísticas, mas se a questão que não tiver texto livre então não são mostradas as respostas dadas por tipo de relação.

Esta página dos relatórios estatísticos é construída com base num gerador de código. O relatório estatístico a visualizar está associado a um processo de avaliação e a uma avaliação. O gerador, com base nestes dados, gera as estatísticas para cada questão, dependendo dos tipos de questão associados à avaliação.

Os resultados para cada tipo de estatística são calculados com base em *Stored Procedures*. Como a classe (*Code-Behind*⁵), que contem o gerador, invoca cada uma das classes que constroem os vários tipos de estatísticas, podem ser criadas mais estatísticas, bastando criar a classe que calcula a estatística e modificar a classe que contem o gerador para passar a invocar devidamente a nova classe.

3.2.2.3.3 Funcionalidades gerais

- **Multi-linguagem** – Toda a aplicação contém um sistema de multi-linguagem incorporado, as linguagens possíveis são português e inglês, sendo possível adicionar outras.

Uma vez que o utilizador inicie a sessão, é identificada a linguagem que será usada. O sistema baseia-se na variável de sessão “*Language*” ou no caso de o

⁵ *Code-Behind* – Paradigma utilizado para programação ASP.NET, que representa o código, mais propriamente as acções que a página e os controles terão. As linguagens de programação usadas poderão ser qualquer uma das que a .NET Framework suporte (C#, VB.NET, J#, qualquer outra linguagem .NET).

utilizador ter seleccionado uma linguagem pelo painel de linguagem, interpreta o texto no URL e identifica a linguagem seleccionada pelo utilizador. Para cada utilizador poder escolher a linguagem que se adequa melhor a si, a implementação da aplicação baseia-se em ficheiros com a extensão “.resx”⁶. Estes ficheiros, usados em ASP.NET, designados por ficheiros de recursos (*resources*), e que têm um ficheiro XML composto pelas palavras (*strings*) que se pretendem traduzir em diferentes linguagens ou caminhos (*path*) em imagens.

Os ficheiros de recursos contêm um par chave/valor. Cada par é um recurso individual.

Foram desenhados controlos (*WebUserControls*⁷), incorporados na *CG.Framework.Web* do departamento, que para além de terem as propriedades dos controlos que herdam (*System.Web.UI.WebControls*), têm mais uma propriedade designada de “*TexKey*”. A esta propriedade é associado o valor da chave do recurso a utilizar, onde o método que carrega o controlo associa o texto deste ao valor da chave do recurso.

Este sistema já se encontrava implementado, mas existiu a necessidade de conhecê-lo pois tive de criar mais recursos à medida que fui desenvolvendo as páginas.

- **Ajuda** – Uma outra funcionalidade já existente é a página de ajuda, que é uma página HTML que contém ajuda sobre como os utilizadores devem proceder para responder às avaliações.
- **Ordenação nas listas** – Todas as listas existentes são ordenadas por número de empregado/ordem alfabética e estão incorporadas numa *DataGrid* que permite visualizar 5, 10, 20 ou todos os elementos da lista, permitindo assim uma melhor procura do elemento pretendido. Esta funcionalidade também já existia.

Para além das funcionalidades que implementei, quero deixar presente que foi necessário efectuar algumas modificações em todas as páginas *Web* da aplicação, pois

⁶ ASP.NET Web Page Resources Overview – msdn2.microsoft.com/en-us/library/ms227427.aspx

⁷ *WebUserControls* – Trata-se de um controlo que é definido pelo programador, tem a extensão “.ascx” e é muito importante na medida que simplifica a reutilização de componentes de Interface num sistema *Web*.

reestruturou-se o *design* de toda a aplicação e tornou-se mais simples o seu funcionamento.

Como as páginas *Web* são desenvolvidas em dois modos separados, o modo de *design* e o modo de *Code-Behind* é neste modo que se efectuam as maiores alterações visuais. Porque é onde se “constrói” a página, ou seja, onde se encontram os *WebUserControls* e estrutura da página,

Também foi necessário compreender todo o código existente, essencialmente as *Framework's* usadas no departamento e o DAL criado para este projecto (ver secção 3.2 *Framework's* internas).

3.2.2.4 Testes

Antes de iniciar o desenvolvimento de novas funcionalidades e alterações à aplicação, foi-me proposto testar a aplicação existente, para, por um lado, conhecer a aplicação e por outro verificar quais as incoerências e possíveis erros que ocorriam na aplicação.

Elaborei um documento onde listavam as incoerências existentes na aplicação. Por exemplo, ao gravar um processo de avaliação a mensagem de sucesso que deveria surgir no rodapé da página não surgia, ou o título da página que surge no browser não correspondia à realidade. Os erros da aplicação eram muito poucos e não graves.

Depois de elaborado o documento, resolvi a lista de incoerências e de possíveis erros. Este processo levou-me apenas alguns dias, cerca de três dias, não estando previsto no planeamento mas previsto no tempo de adaptação e conhecimento do código já implementado.

Já na fase de desenvolvimento de novas funcionalidades e alterações, sempre que concluía a implementação da nova funcionalidade ou alteração, eu testava a aplicação nessa funcionalidade ou alteração, minimizando assim os possíveis erros.

Depois de concluída toda a fase de desenvolvimento, foi elaborado por mim e com aprovação final do Daniel Vinagre (chefe de equipa) um plano de testes. Neste plano constava para cada funcionalidade o seu objectivo e quais os resultados finais na aplicação. Este plano era entregue a um membro da equipa de suporte técnico ao cliente do departamento das Aplicações Standard da Capgemini, que deveria testar cada uma das funcionalidades da aplicação e reportar os problemas encontrados.

Uma vez testada a aplicação, são resolvidos os problemas encontrados e mais uma vez testados tanto por mim como pelo mesmo elemento da equipa de suporte técnico.

Este processo é repetido até os testes serem concluídos com sucesso.

A primeira versão desta renovação da aplicação passou pela fase de testes e foram reportadas algumas anomalias, principalmente a nível funcional de botões, mudança de páginas nas grelhas de listagens e também o mau funcionamento da sessão no acesso de *back-end* ao administrador. Com a sessão iniciada numa página de acesso por *back-end* (funcionalidades administrativas), no momento em que é fechada a janela do *browser* onde a sessão foi iniciada, a sessão não era finalizada. Permitindo assim a qualquer pessoa que acede-se ao mesmo computador, que não tivesse conhecimento da *password* de acesso à aplicação e antes do tempo de sessão expirar, aceder a todas as funcionalidades de *back-end*, porque não necessitava de iniciar uma nova sessão.

Estes problemas foram resolvidos e testados novamente por mim e pela mesma pessoa da equipa de suporte técnico, não sendo reportado mais nenhum problema.

3.2.3 Fase 3 – SigCRM (Sistema Integrado de Gestão – *Customer Relationship Management*)

Tal como já foi referido o segundo projecto passou a ser o SigCRM, substituindo o projecto estabelecido inicialmente, o Portal do Empregado.

O projecto SigCRM foi integrado numa equipa de cinco elementos, incluindo o chefe de equipa Daniel Vinagre.

No momento da minha integração no projecto, a fase de análise e modelo de dados já estava concluída. Por isso apenas implementei as funcionalidades que me foram atribuídas e efectuei os testes do meu trabalho desenvolvido e quando necessário, efectuei alterações ao modelo de dados.

Ao longo do decorrer da implementação do projecto e apesar do seu desenvolvimento ter sido distribuído pelos elementos da equipa, nunca deixou de existir comunicação entre todos os elementos, pois existiam as reuniões diárias (*Scrum*) e ainda a necessidade de comunicar entre todos aquando da ocorrência de alguma dificuldade, de discutir ideias, para tornar uma determinada funcionalidade mais simples, ou mesmo porque determinada funcionalidade tem de ser coerente com outra.

Outro facto é que para quatro dos cinco elementos da equipa, a maioria das linguagens de programação e ferramentas utilizadas eram pouco conhecidas, por isso a comunicação tornou-se importante no aspecto de se necessitar de mais apoio para o desenvolvimento correcto e cumprimento dos prazos de entrega previstos.

3.2.3.1 Funcionalidades

Como já foi referido, esta aplicação é uma aplicação *Web*, foi toda implementada com ASP.NET, C#.NET e quando necessário JavaScript, com o auxílio da ferramenta Microsoft Visual Studio .NET e Microsoft SQL Server e Framework's da Capgemini e .NET.

De seguida descrevo resumidamente algumas funcionalidades gerais que não foram implementadas por mim, mas para as quais foi necessário entender como foram desenvolvidas. É que todas as páginas *Web* da aplicação têm a mesma estrutura e para não se implementar conceitos que já foram implementados noutras páginas, todas as páginas são desenvolvidas com base na herança. Por isso é necessário conhecer algumas funcionalidades.

3.2.3.1.1 Funcionalidades gerais

- **Acessos/permisões e sessões** – Tal como a aplicação SigAV360, o SigCRM tem um sistema de acessos a utilizadores e tempo de sessão para cada utilizador.

O sistema que o SigCRM adopta é o uso de *cookies* (secção 3.2.2.2) para manter a sessão do utilizador.

Optou-se pelo uso de *cookies* pois estes tornam-se mais seguros que o modo *cookieless*, devido a neste último o identificador da sessão ser passado no URL. Mesmo sendo cifrado, é pouco seguro dado que o identificador é visível na rede, permitindo facilmente a um *hacker*⁸ obter este valor. O artigo de Bryan Sullivan intitulado “*Top 10 Application Security Vulnerabilities in Web.config Files* -

⁸ Hacker – <http://pt.wikipedia.org/wiki/Hacker> – pessoa que pretende danificar dados ou aceder a informações confidenciais para fins maliciosos.

*Part One - Cookieless Session State Enabled*⁹ explica melhor esta vulnerabilidade.

Ainda no sistema de sessão, o SigCRM usa um *cookie* criado no ficheiro de “.config¹⁰”. Este *cookie* foi criado para poder controlar a sessão no caso em que o utilizador feche a janela do *browser* a sessão mantém-se, permitindo a este, dentro do tempo de sessão definido, voltar a entrar na aplicação sem efectuar novamente o *login*. Para isto basta o utilizador escolher no *login* a opção de “lembrar na próxima vez”. A forma de implementação baseia-se numa tabela existente na base de dados, que controla quais as sessões activas e qual o identificador da sessão (“*SessionId*”).

Os *cookies* referentes ao utilizador estão cifrados por razões de segurança.

A aplicação tem um *login* por utilizador e outro, *standard* e configurável, de administrador que tem acesso a todas as funcionalidades.

Uma vez efectuado um *login* com sucesso é iniciada uma sessão com um *time-out* de 60 minutos. Esta configuração de tempo de sessão é feita no ficheiro “.config”, colocando no formato XML interpretado pelo sistema a configuração do tempo de sessão. O SigCRM usa esta configuração para o tempo de sessão e uso de *cookies*:

```
<authentication mode="Forms">

<forms loginUrl="Login.aspx" protection="All" timeout="60"
name="CRMCookie" path="/" requireSSL="false"
slidingExpiration="true" defaultUrl="default.aspx"
cookieless="UseCookies" enableCrossAppRedirects="false" />

</authentication>
```

Figura 6 – Configuração de Cookies no ficheiro “. config”

A forma de controlar se uma sessão expirou ou não, é feita por herança. Existe uma classe base de que todas as páginas da aplicação (*Code-Behind*) devem

⁹ Artigo de Bryan Sullivan - <http://www.developerfusion.co.uk/show/6678/6/>

¹⁰ “.config” - <http://en.wikipedia.org/wiki/Web.config> - ficheiro configuração de aplicações .NET que permite indicar por exemplo as *connectioString* que a aplicação deve de usar para aceder a uma base de dados, ou indicar modos de autentificação de páginas *Web*, entre outros menos relevantes para o caso.

herdar (“*PageBase*”). É nesta classe, no método *Load*, que é verificado se o tempo de sessão terminou e se assim for é necessário redireccionar o utilizador para a página de *login*. Caso não tenha expirado o tempo de sessão o valor é actualizado.

Esta classe também trata dos acessos às páginas permitidas para cada utilizador. Um utilizador pertence a um ou mais grupos de permissões, e cada grupo tem um ou mais acessos a páginas.

No desenvolvimento das páginas e dependendo do tipo de página (ver tópico seguinte sobre “Estruturas páginas *Web*”), estas devem de efectuar um *override*¹¹ a um método. Ao carregar a página, a instância, verifica qual é a permissão necessária para visualizar a página e com o *cookie* (“*UserId*”, informação referente ao utilizador), verifica se o utilizador tem acesso à página em questão.

Tanto os utilizadores como as permissões que cada um tem estão mantidos na base de dados da aplicação. E a sua manutenção é efectuada na aplicação por um utilizador com acessos administrativos.

- **Estruturas de páginas *Web*** – Todas as páginas têm a mesma estrutura, o menu vertical do lado direito, no canto superior o logótipo da empresa e nome da aplicação e no centro o conteúdo de cada página.

Para evitar a repetição de código, foi criada uma classe base, designada “*PageBase*”, esta classe herda do *System.Web.UI.Page* e tal como já referi atrás, trata dos tempos de sessão e acessos às páginas pelas permissões que cada utilizador tem e para ainda das mensagens de erro/sucesso que surgem numa barra de estado no cabeçalho de cada página.

Depois existem dois tipos de páginas, as páginas de listagens, que contêm sempre uma pesquisa com filtros definidos para a página e uma listagem com os resultados da pesquisa; e as páginas com a informação de um conceito existente na aplicação, por exemplo os dados de uma entidade.

¹¹ *override* - [http://msdn2.microsoft.com/en-us/library/ebca9ah3\(VS.71\).aspx](http://msdn2.microsoft.com/en-us/library/ebca9ah3(VS.71).aspx) – Forma como as classes que herdem de uma classe base podem modificar propriedades/metódos/eventos da classe base.

Todas as páginas *Web* têm um controlo base designado “*PageHeader*”, este controlo tem como base os botões de voltar, gravar, apagar, inserir e exportar, dependendo do tipo de página e, sendo possível no *Code-Behind* das páginas *Web*, adicionar outros botões. Assim estas apenas têm de incluir o controlo e não necessitam de se preocupar com os eventos dos botões.

Para criar as páginas do tipo de listagens, existe uma classe base da qual devem herdar, designada “*ListPageBase*”. Esta classe base herda por sua vez da classe base “*PageBase*”, já falada.

Na classe “*ListPageBase*”, estão implementados todos os métodos públicos que as páginas *Web* podem necessitar e “obriga-as” a modificar (*override*) certos métodos que a classe base necessita para poder carregar automaticamente os seus controlos na página *Web*. Assim informação como o título da página, permissões, o evento “redireccionamento do botão voltar” e sobre que objecto do DAL se efectua as pesquisas, é tudo tratado na classe base, não sendo da preocupação de quem implementa a página e evitando a repetição dos métodos que tratam desta informação, facilitando possíveis alterações a estes (os métodos encontram-se todos numa classe e não espalhados pelas páginas *Web*).

As páginas *Web* que representem um conceito da aplicação devem herdar de uma classe base designada de “*RecordPageBase*”. Esta classe base também herda da classe “*PageBase*”, e trata de informação como título das páginas, permissões, modo de acesso (inserção, modificação, leitura ou duplicação de um registo), o evento de redireccionamento do botão voltar, eventos dos botões gravar e apagar, acções a tomar depois de gravar dados, mensagem de erro/sucesso e sobre que registo do tipo de um objecto DAL se efectua as acções de gravar, apagar, modificar ou inserir e ainda as validações dos controlos das páginas.

Esta classe baseia-se nos dados contidos no *URL encoding* (codificação do URL)¹² e *cookies*, em que dados como a informação do utilizador, registo actual de um objecto DAL, e informação da página anterior, podem ser recolhidos.

A figura 7 representa as heranças das classes base que acabei de mencionar.

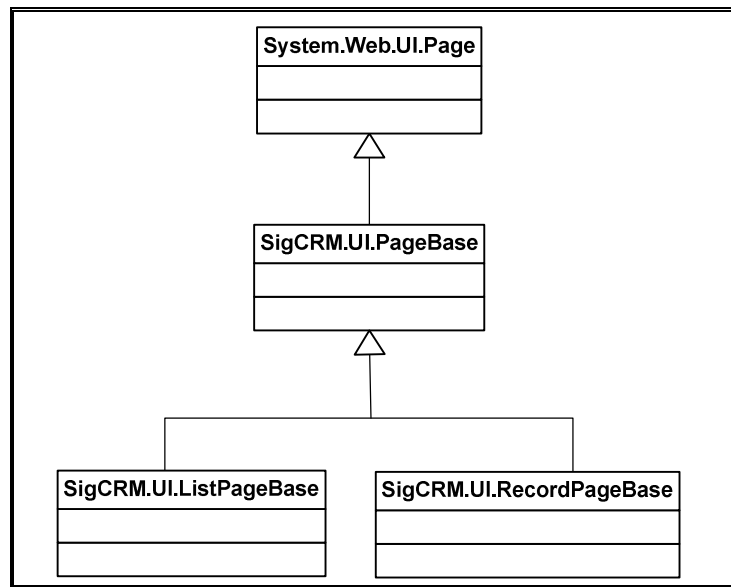


Figura 7 – Diagrama de classes base usadas nas páginas Web da aplicação SigCRM

- **URL encoding** – As páginas Web usam conceito *URL encoding* para passar a informação necessária de página para página.

Em páginas que representem um conceito da aplicação (exemplo: dados da entidade, sendo do tipo “*RecordPageBase*”), é passado no URL, dados como o identificador único do registo, o modo de acesso (inserção, modificação, leitura ou *clone* do registo) entre outros necessários, mas o mais essencial um par atributo/valor designado “*BackLocation*” que tem como valor o endereço da página de retorno com respectivo com *URL encoding*, ou seja, é uma forma de

¹² *URL encoding* – Permite guardar informação no endereço URL, a seguir ao endereço da página. Esta informação é armazenada numa *query string* que é adicionada ao fim do endereço URL. Exemplo do SigCRM:

Tarefa.aspx?Action=UPDATE&Id=1800&BackLocation=Sa2aUFH6b0lhtC XKDFP07GoGFAhmdFZp134GbSdeJAQ=,

Onde “*Action*”, “*Id*” e “*BackLocation*” são nomes dos atributos existentes na *query string* (seguido do sinal “=” encontra-se o respectivo valor).

manter um histórico de onde o utilizador passou. O valor deste par atributo/valor encontra-se cifrado por razões de segurança. Mais uma vez quem trata de decifrar este valor é a classe base “*RecordPageBase*” e redirecciona o utilizador para a página anterior e em casos que sejam páginas do tipo de listagem (“*ListPageBase*”), carrega os filtros de pesquisa com os dados da pesquisa efectuada.

3.2.3.1.2 Funcionalidades implementadas

Nesta secção descreverei as funcionalidades implementadas por mim na aplicação SigCRM. Inicialmente vou descrever os conceitos de CRM que desenvolvi, (páginas do tipo “*RecordPageBase*”), pois têm uma estrutura muito idêntica, diferindo apenas nas funcionalidades que cada conceito contém. Depois descreverei as restantes funcionalidades que não façam parte de conceitos de CRM.

3.2.3.1.2.1 Funcionalidades de conceitos de CRM

Para cada um destes conceitos de CRM que desenvolvi, tal como já mencionei, foram implementadas os dois tipos de páginas, “*ListPageBase*” e “*PageBase*”. Como para as páginas do tipo listagem (“*ListPageBase*”), a estrutura é igual para todos os conceitos de CRM, não vou descrever para cada um a implementação deste tipo de página. No fim desta secção explicarei as funcionalidades gerais que uma página deste tipo tem. Por isso vou apenas descrever as páginas que contém a informação relativa a um registo (“*RecordPageBase*”).

- **Oportunidades** – Conceito que representa uma oportunidade de venda a uma entidade (por exemplo um cliente de uma empresa). Através desta funcionalidade pode-se controlar todos os aspectos relacionados com uma eventual venda relacionada com a entidade.

É possível associar uma oportunidade de venda a uma entidade de duas formas. A partir da consulta da entidade pretendida, ou pela própria funcionalidade das oportunidades.

Para associar a uma oportunidade a entidade pretendida, a funcionalidade tem um controlo que permite efectuar uma pesquisa da entidade.

A figura 8 ilustra o controlo que permite a pesquisa de entidades, o botão idêntico a uma lupa permite pesquisar a entidade, mas caso se saiba o número da entidade basta inseri-lo na caixa de texto activa e assim que esta perca o *focus*, um evento do controlo tenta encontrar a entidade com o respectivo número, caso a encontre associa à caixa de texto inactiva o nome da entidade e permite aceder a esta a partir de um botão, caso contrário coloca uma mensagem a indicar que o registo não foi encontrado.

Para este tipo de controlo ter este comportamento, existe uma classe base designada de “*LOVControlBase*” que o controlo deve herdar. Por esta razão o controlo é do tipo LOV (*List of Values*) que indica que este herda da classe base.

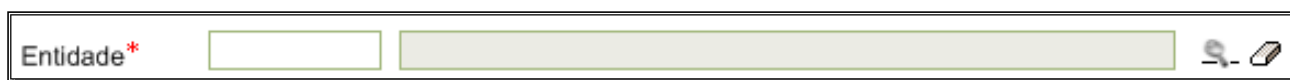


Figura 8 – Controlo de pesquisa de entidades

A oportunidade de venda para além da entidade a associar tem outros dados como a fase da oportunidade, o utilizador responsável, o utilizador que criou a oportunidade, o tipo de contacto que deu origem à oportunidade, a probabilidade de se concretizar a venda, a fase da proposta na entidade, a pessoa da entidade a contactar, os dados de facturação e adjudicação, e ainda uma funcionalidade opcional de numeração automática da proposta.

Para implementar a numeração automática da proposta, e como existia uma forte possibilidade de outras funcionalidades necessitassem de usar a numeração automática, desenvolvi uma funcionalidade designada “Numeração Automática” e que por defeito apenas os utilizadores com direitos administrativos podem configurar, portanto apenas está disponível no menu administração. Falarei de como implementei esta funcionalidade mais à frente.

Uma vez criada a oportunidade de venda, é possível associar produtos, processos, tarefas, contactos, relatórios de visita, documentos e observações à oportunidade. Estas funcionalidades surgem no fundo da página *Web* sobre a forma de *tabs* (separadores), a figura 9 representa estas funcionalidades e foram implementadas pelo colega Rui Rodrigues.

Estas *tabs* são controlos adicionados à página, são controlos do tipo LOV, por isso herdam da classe base “*LOVControlBase*”. Na figura 9, o que representa o controlo do tipo LOV, também implementado pelo meu colega, é o que se encontra marcado com a linha a vermelho.

Para o exemplo mostrado, “associar processo”, o LOV controlo permite inserir um novo processo (a um processo é possível associar uma entidade e uma oportunidade já existente), ou então associar a um processo já existente, que faz com que apareça uma janela de pesquisa do processo, permitindo seleccionar o pretendido.

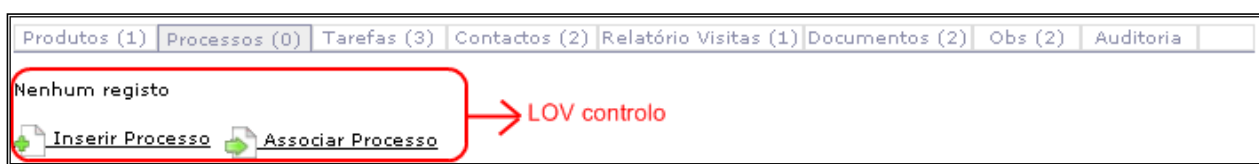


Figura 9 – Tabs existentes na oportunidade

No cabeçalho da página da oportunidade é possível efectuar as acções básicas como gravar, eliminar, duplicar e voltar. Mas para esta funcionalidade de criar oportunidades de vendas foi acrescentada mais uma acção ao cabeçalho: consultar o cronograma com os diferentes estados pelo qual a oportunidade passou.

Este cronograma é um gráfico de barras que abre numa janela *pop-up*, os dados que devem surgir para cada barra do gráfico são adquiridos a partir de uma *view* existente na base de dados. Esta é construída com base nas tabelas que contêm os contactos, as tarefas associadas à oportunidade e as alterações (registo de auditoria, é uma tabela de auditorias para a oportunidade) efectuadas. O gráfico mostra em cada barra uma descrição com uma imagem e ainda a data do estado da oportunidade.

- **Objectivos de Venda** – Permite definir os objectivos de venda por mês. Permite ter uma visão global dos valores/percentagens atingidos e não atingidos para cada mês, por consulta de valores/gráficos.

Para esta funcionalidade é possível inserir um objectivo de venda por mês e indicar ou não um período de facturação, já que para algumas empresas, o período de facturação não é igual ao de início/fim do mês.

Uma vez inseridos os dados referentes ao objectivo de venda, é efectuado o cálculo, com base nas datas previstas de adjudicação e facturação das oportunidades de venda, da percentagem realizada, dos totais facturados/adjudicados e por realizar para o objectivo estabelecido.

São também disponibilizados duas listagens, uma das oportunidades de venda fechadas com sucesso e outra das oportunidades não fechadas no período do objectivo. Estas duas listas são do tipo *UserControlWeb GridView* e têm em cada registo delas uma ligação para a oportunidade e para a entidade, permitindo assim uma melhor análise das oportunidades e consulta das entidades em causa.

No cabeçalho são apresentados os botões gerais, eliminar, gravar e voltar.

Uma diferença que surge neste conceito é que a página do tipo listagem (“*ListPageBase*”) permite aceder a duas páginas: à página com informação do objectivo de venda no período seleccionado (faz parte das funcionalidades gerais para as páginas “*ListPageBase*”) e ainda a uma página que permite consultar dois gráficos com os objectivos de venda em valores e outro em percentagens de um determinado ano, separado por mês. Estes gráficos são páginas que foram desenvolvidas e depois adicionados na página que mostra os gráficos como imagens.

Para construir este gráficos foi usado um programa de código livre de um controlo que permite construir gráficos de vários tipos. Este controlo designa-se de *ZedGraphWeb*¹³. Para cada gráfico é indicado o tipo de gráfico a construir, parametrizado os eixos e atribuídos os resultados.

Para incluir este controlo na aplicação apenas foi necessário adicionar o ficheiro “.dll”¹⁴ às pastas *bin* do projecto SigCRM (a *Web* que compõe a aplicação). Para usar o controlo deve fazer-se a sua invocação numa página ou controlo tal como se efectua a invocação dos *WebUserControl*’s. E no *Code-behind* da página ou controlo implementa-se o código que parametriza e constrói o gráfico.

¹³ *ZedGraphWeb* - http://zedgraph.org/wiki/index.php?title=Main_Page

¹⁴ “.dll” (*Dynamic-link library*) - Biblioteca de ligação dinâmica, é a implementação feita pela Microsoft para o conceito de bibliotecas compartilhadas nos sistemas operativos Microsoft Windows e OS/2.

Em relação às páginas do tipo listagem (“*ListPageBase*”) destes conceitos que descrevi e tal como já referi, estas têm a mesma estrutura. São compostas por um controlo do *WebUserControl GridViewExtensionPanel*¹⁵, que faz parte da *Framework* da Capgemini (CG.Framework.Web.UI) e um *WebUserControl GridView*.

Os dois controlos não foram implementados por mim, mas houve a necessidade de conhecer estes controlos para saber o que devo de implementar a nível de programação de forma a poder obter os resultados pretendidos.

Para o *GridViewExtensionPanel* preciso de indicar o controlo de pesquisa, as imagens para os botões que surgem no cabeçalho da página e quais são os botões que devem de aparecer, quantos elementos surgem na lista da *GridView* e ainda sobre que objecto do DAL é baseada a pesquisa.

Para a *GridView*, que tem de ser colocada na definição do *GridViewExtensionPanel*, apenas é necessário adicionar-lhe as colunas.

O controlo de pesquisa que é associado ao *GridViewExtensionPanel* deve de ser também desenvolvido. Basta criar um controlo (extensão “.ascx”) e indicar que este herda de uma classe base, “*SearchControlBase*”, existente na *Framework* da Capgemini (CG.Framework.Web.UI.BaseClasses), também um *WebUserControl* (herda do System.Web.UserControl), composta por métodos que permitem efectuar uma pesquisa sobre um objecto do DAL.

Foi preciso desenvolver os controlos de pesquisa necessários. No *design* dos controlos, coloco os controlos pretendidos para a pesquisa (serão os filtros de pesquisa) e no *Code-Behind* modifico o método da classe base que retorna a *query* para a pesquisa sobre o objecto DAL indicado no *GridViewExtensionPanel*.

Uma vez desenvolvidas as páginas do tipo listagem (“*ListPageBase*”) quem trata das acções destas são as classes base, *GridViewExtensionPanel* e “*SearchControlBase*”.

¹⁵ *WebUserControl GridViewExtensionPanel* – Controlo genérico criado para usar na aplicação, *standard* para todos os membros da equipa usarem exactamente o mesmo controlo com as mesmas funcionalidades, controlo de pesquisa em que apresenta os resultados numa *grelha*.

3.2.3.1.2.2 Outras funcionalidades

- **Numeração automática** – A funcionalidade de numeração automática já foi referida na funcionalidade de oportunidades.

Devido à possibilidade de outras implementações de conceitos CRM virem a necessitar de ter numeração automática, tomou-se a decisão de colocar esta funcionalidade independente da oportunidade.

A sua implementação passa por criar duas tabelas na base de dados, uma que tenha as numerações possíveis na aplicação (uma descrição, máscara de numeração e valor – *string* que indica para que funcionalidade usa a máscara) e outra que contenha a *query* em SQL que devolva a numeração seguinte para a máscara.

Foi necessário desenvolver as páginas que permitem efectuar as manutenções destas tabelas (para cada tabela duas páginas, uma do tipo “*ListPageBase*” e outra do tipo “*RecordPageBase*”).

Esta funcionalidade só está acessível pelo menu da administração.

Nas páginas que necessitem de usar esta funcionalidade, a sua implementação baseou-se em colocar os *WebUserControl CheckBox* e *DropDownList*, do tipo AJAX¹⁶, e colocar apenas visível o *WebUserControl CheckBox*, este tem associado um evento que ao ser desencadeado acede à tabela de numeração que contém as máscaras com valor da funcionalidade em questão e mostra-as na *WebUserControl DropDownList* (já visível). Para a máscara seleccionada na *DropDownList* (para isso é necessário associar um evento a este controlo que efectue os cálculos) é calculado o valor do próximo registo e disponibilizado no

¹⁶ AJAX – (acrónimo em língua inglesa de *Asynchronous Javascript And XML*) Técnica de desenvolvimento *Web* usada para criar páginas *Web* interactivas. A intenção é de fazer com que as páginas sejam mais rápidas por enviarem pequenas quantidades de informação ao invés da página toda para o servidor. Assim evita que a página inteira seja carregada sempre que o utilizador efectue um acção que necessite de efectuar alterações à página. O objectivo final é aumentar a capacidade de resposta, interactividade, funcionalidade e usabilidade da página.

lugar do campo de numeração automática, dando a hipótese ao utilizador de modificar esta numeração calculada.

- **Fundir entidades** – Para entender este conceito, imagine-se o seguinte exemplo: dois utilizadores criam duas entidades com nomes idênticos, por exemplo, FCUL e Faculdade de Ciências da Universidade de Lisboa, (caso fossem iguais a aplicação notificava ao segundo utilizador no momento da criação que já existe uma entidade com o mesmo nome), individualmente associam oportunidades, tarefas, produtos e relatórios de visita para a entidade respectiva. Até que em conferência na hora da pausa para o café, descobrem que têm a mesma entidade em comum. Para solucionar este problema bastava-lhes fundir as duas entidades numa só, mantendo toda a informação já existente. Ou seja, ter apenas uma entidade e ter associada a ela os relatórios de visita, as oportunidades, as tarefas e produtos que estavam associados às duas entidades idênticas.

Para estes e outros possíveis problemas que ocorram em que a solução se baseie na fusão de entidades, foi desenvolvida esta funcionalidade, acessível no menu de administração.

Para o utilizador a funcionalidade é apresentada com dois controlos do tipo LOV, já mencionados, que permitem pesquisar entidades. O utilizador indica as duas entidades a fundir (entidade 1 e entidade 2) e executa a acção de fundir (funde a entidade 1 na entidade 2), uma vez concluído o processo, o utilizador é reencaminhado para o registo da entidade final (será a entidade 2 mais os dados da entidade 1).

O desenvolvimento desta funcionalidade baseia-se em implementar uma *Stored Procedure* em SQL, que com base nos identificadores únicos das duas entidades a fundir, modifica os registos das tarefas, contactos, relatórios de visita, moradas, documentos, etc., da entidade 1, ou seja, modifica todos registos associados à entidade 1 para passarem a associar a entidade 2 (em SQL, modifica-se os valores das chaves estrangeiras nas tabelas que referenciem a entidade 1 para a entidade 2). E uma vez desencadeado o evento associado à acção fundir é executada esta *Stored Procedure* e apagada a entidade 1. De seguida redirecciona-se para página que representa o registo da entidade 2.

- **Ligação ao módulo SigGC** – A aplicação SigCRM está ligada à aplicação SigGC permitindo efectuar importações e exportações de entidades, importar produtos, sincronizar contactos e moradas de entidades e visualizar documentos em aberto de uma entidade (documentos por liquidar).

Cada uma das funcionalidades necessita de conhecer a localização do servidor e nome da base dados da aplicação SigGC. Como esta localização pode mudar ou pode não existir (o cliente que adquire o SigCRM pode não ter adquirido o SigGC), é necessário estas configurações estarem num local acessível. Por isso decidiu-se colocar estas configurações numa tabela, entre outros registos, esta continha a *ConnectionString*¹⁷ da localização do SigGC e se a aplicação usa esta funcionalidade.

- **Importar e exportar:** As funcionalidades de importar e exportar são desenvolvidas como páginas do tipo listagem (“*ListPageBase*”). No desenvolvimento das páginas constrói-se o *DataSet* que ficará associado à *GridView* da página. Este *DataSet*¹⁸ é preenchido com base na *ConnectionString* da localização do SigGC e numa *query* para o tipo de funcionalidade (consta também na tabela de configurações). É sobre este *DataSet* que são efectuadas as pesquisas (o controlo de pesquisa – LOV – efectua pesquisas sobre o *DataSet*).

Para estas funcionalidades é acrescentado ao *header* das páginas botões que permitem seleccionar todos, nenhum e importar. As páginas devem de implementar os eventos destes botões para obter o resultado pretendido.

- **Sincronizar:** Na página de uma entidade existe um controlo (*CheckBox*) que indica se a entidade irá sincronizar com a SigGC, tal como nas moradas e contactos da entidade, ou seja, caso a entidade tenha a opção seleccionada, todos os registos de moradas e contactos que tenham esta opção seleccionada são actualizados/modificados pelos dados que constam na SigGC.

¹⁷ *ConnectionString* – Uma *string* que contém a informação necessária para ligar a uma base de dados. Inclui informação como o nome da base de dados e o servidor, pode conter informação de autenticação com o utilizador e *password*.

¹⁸ *DataSet* – Objecto ADO.NET que permite aceder a dados de uma base de dados. Um *DataSet* permite conter um ou mais *DataTable* (tabela com dados da base de dados). Permite que uma aplicação acesse e modifique os dados da base dados.

Para esta funcionalidade apenas é desenvolvido os controlos das páginas que indicam se se pretende sincronizar os dados com a SigGC. Depois juntamente com a instalação da aplicação no cliente é instalado um serviço, que é executado num determinado intervalo de tempo, configurável.

Este serviço foi implementado pelo chefe de equipa, Daniel Vinagre, pois este também é responsável pelo envio dos correios electrónicos da aplicação, (alertas da aplicação, como por exemplo tarefas/oportunidades em atraso, atribuídas ou modificadas e listagens de oportunidades/tarefas/objectivos de venda para utilizadores específicos) mas foi da minha responsabilidade implementar o método que efectua a sincronização.

Este método baseia-se em *Stored Procedures* criadas na base de dados, que retorna todas as entidades que são sincronizáveis, depois para cada uma destas é retornado todas as moradas e contactos que são para sincronizar e estes são substituídos pelos os dados que constam na SigGC. Os dados que constam na SigGC e não constam no SigCRM são criados na entidade e indicados como sincronizáveis.

A implementação deste método passa por identificar a *ConnectionString* da SigGC e depois efectuar as alterações necessárias do lado do SigCRM.

- Documentos em aberto: Esta funcionalidade está disponível a partir de uma entidade caso a aplicação use a funcionalidade de ligar à SigGC.

Permite consultar os documentos em aberto e respectivo valor, data de vencimento, data do documento e saldo. Estas informações surgem numa listagem ("*ListPageBase*" mas sem controlo de pesquisa) e no fim da listagem os valores totais em aberto, vencidos e total (vencidos + em aberto).

A implementação desta funcionalidade consiste numa página de listagem, em que o *DataSet* da *GridView* é preenchido com base na *ConnectionString* da localização da SigGC e ainda uma *query* construída no *Code-Behind* da página.

- **Controlo de HTML** – A aplicação SigCRM tem uma funcionalidade de *MailingList* e o conceito de contacto (qualquer tipo de contacto efectuado ao cliente). Para estes e principalmente para a funcionalidade de *MailingList*, existe uma necessidade evidente de se poder usar texto em HTML.

Como um controlo deste tipo é bastante complexo de implementar e implica o desenvolvimento de muito código em *JavaScript* e sabendo da existência de um editor HTML para ASP.NET, um programa de código livre bastante utilizado, designado *FreeTextBox*¹⁹, decidiu-se usá-lo na aplicação.

Este controlo é compatível com os *browsers Internet Explorer, Mozilla e Firefox*, o que tem sido uma grande preocupação nossa, já que a maioria dos clientes usam *Internet Explorer*, mas temos conhecimento de alguns que usam outros *browsers* como o *Firefox*, nós garantimos a quase totalidade de uso da aplicação em *Firefox* e totalidade em *Internet Explorer*.

Uma vez adicionado a pasta com o código *JavaScript*, imagens entre outros conteúdos do *FreeTextBox* e adicionada o ficheiro com extensão “.dll” à pasta “bin” do projecto SigCRM (à *Web* que compõe a aplicação), a invocação do controlo *FreeTextBox* pode ser efectuada numa página ou *WebUserControl* (“.aspx” e “.ascx”), como se fosse um controlo usual.

Na funcionalidade de *MailingList* o controlo é adicionado à página e no *Code-Behind* formatou-se o controlo (disponibilizou-se apenas as funcionalidades pretendidas), para além disto, implementou-se uma forma de permitir ao utilizador escolher se pretende o envio do texto do correio electrónico em HTML ou texto simples. Caso opte por correio electrónico com HTML o controlo *FreeTextBox* é disponibilizado, caso contrário apenas é disponibilizado uma caixa de texto normal.

Algumas tabelas da base de dados da aplicação tiveram de ser modificadas para poderem armazenar texto em HTML.

Como o uso da *MailingList* armazena nos contactos da entidade o envio de um contacto tipo correio electrónico, também foi necessário implementar o controlo no conceito de contactos.

¹⁹ *FreeTextBox* - <http://freetextbox.com/default.aspx>

3.2.3.2 Testes

Os testes da aplicação SigCRM seguiram o mesmo método já mencionado na aplicação SigAV360. Todas as funcionalidades implementadas por mim foram testadas e depois elaborado um documento com os testes a efectuar para estas. Uma vez aprovado pelo chefe de equipa, Daniel Vinagre é redireccionado para um elemento da equipa de suporte técnico ao cliente, do departamento das Aplicações Standard da Capgemini, que testou cada uma das funcionalidades e reportou os problemas/incoerências encontrados. Estes são resolvidos e novamente testados pela mesma pessoa, repetido o processo de testes até a aplicação se encontrar sem problemas/incoerências.

Para esta aplicação apenas me foram reportados os erros encontrados nas funcionalidades que desenvolvi. Os erros encontrados nos primeiros testes foram relacionados com a numeração automática e objectivos de venda.

Na numeração automática algumas das *query's* associadas às máscaras de numeração não tinham o comportamento correcto, estavam mal construídas. Foi necessário modifica-las e alterar os registos destas na base de dados. Nos objectivos de venda, os cálculos efectuados para os gráficos estavam mal feitos. A *view* criada para um dos gráficos efectuava mal os cálculos para cada mês do ano.

Os restantes problemas encontrados deviam-se a controlos que não funcionavam bem, ou porque não lhes era atribuído um valor, nos casos de visualização ou modificação de um registo (por exemplo, o controlo do estado da entidade não estava preenchido, quando o registo na base de dados tinha um estado associado), ou porque a funcionalidade de duplicação deveria de colocar alguns controlos por defeito com um determinado valor e não o faziam (por exemplo, numa duplicação de uma tarefa, o estado de aprovação deveria de estar associado a “a aguardar aprovação” e o estado da tarefa deveria de ficar associado com o valor de “pendente”), entre outros relacionados com os comportamentos dos controlos.

Estes problemas foram corrigidos, testados por mim e pela mesma pessoa da equipa técnica que não encontrou mais erros para as funcionalidades desenvolvidas por mim.

A aplicação passou a ser utilizada internamente do departamento de Aplicações Standard da Capgemini, o que irá ajudar a encontrar possíveis erros na aplicação e novas funcionalidades.

Capítulo 4

Conclusão

Neste capítulo é descrito um sumário do trabalho realizado no âmbito do PEI, um comentário crítico e ainda o trabalho futuro que se pode realizar nas aplicações em que participei.

4.1 Sumário

Este relatório descreve todo o trabalho realizado no âmbito PEI (Projecto de Engenharia Informática) na empresa Capgemini, Portugal. Este trabalho teve a duração de 9 meses, e engloba duas aplicações, o SigAV360 e SigCRM.

Para a aplicação SigAV360, o trabalho realizado passou por melhorar a aplicação, uma vez que esta já estava desenvolvida, mas para um cliente específico, por isso demasiado específica. Foi necessário tornar a aplicação *standard* e para isso foram desenvolvidas novas funcionalidades, melhorou-se o sistema de navegação e o aspecto visual da aplicação.

Para realizar este trabalho foi necessário efectuar modificações no modelo de dados da aplicação, entender todo o código desenvolvido, organizar a estrutura da aplicação e desenvolver novas funcionalidades sendo as principais implementar um sistema de navegação autónomo, independente do *browser*, a par da inclusão de pontos de referência relativos às páginas de navegação, consultar relatórios estatísticos e a ligar à aplicação SigGC afim de importar empregados.

Para a aplicação SigCRM, integrada numa equipa e em fase de criação da aplicação, foram-me atribuídas algumas funcionalidades. As mais relevantes foram a numeração automática de alguns conceitos de CRM, permitir fundir duas entidades, permitir usar texto HTML em algumas funcionalidades, visualização de gráficos de objectivos de venda e a ligação à aplicação SigGC para permitir a aplicação importar e exportar entidades, importar produtos, sincronizar entidades, moradas e contactos com a SigGC e consultar os documentos em aberto de uma entidade.

4.2 Comentário crítico

Durante os 9 meses que decorreram no âmbito do PEI e principalmente nos primeiros meses, deparei-me com algumas dificuldades em compreender certos conceitos de negócio das aplicações onde estive integrada. Foi necessário adquirir esses conhecimentos questionando os colegas de trabalho e ainda recorrendo a pesquisas bibliográficas.

Sobre a linguagem de programação e ferramentas utilizadas, deparei-me apenas com as dificuldades normais de ambientação, tendo sempre apoio incondicional do meu chefe de equipa, Daniel Vinagre, e de qualquer outro colega do departamento a quem me dirigisse.

Uma crítica que posso ter em relação ao trabalho que realizei na primeira aplicação, SigAV360, foi na minha opinião, que o departamento de Aplicações Standard da Capgemini, ao desenvolver a aplicação teve uma abordagem errada, esta deveria ter sido elaborada o mais *standard* possível, apesar de ter sido criada pelo contacto de um cliente.

Deveria ter-se efectuado o estudo inicial, contactar a população alvo da aplicação, o departamento de recursos humanos de alguns clientes. Desta forma as futuras alterações à aplicação seriam menos profundas, não seria necessário reestruturar a aplicação e modelo de dados por inteiro.

Sobre a metodologia de testes utilizada no departamento, também não posso concordar por inteiro. Sou da opinião de que os testes devem de ser efectuados numa primeira fase pelos elementos que desenvolvem a aplicação e numa segunda fase por um elemento neutro ao desenvolvimento da aplicação. Mas não concordo que uma vez que sejam detectados os erros e corrigidos que seja o mesmo elemento neutro teste novamente as funcionalidades. É evidente que existe uma possibilidade de não serem detectados novos erros que possam ter sido originados devido à correcção dos primeiros, uma vez que este elemento testará a aplicação menos profundamente que a primeira vez. Deveria de existir uma equipa de testes e para esta equipa ser criado um plano de testes.

Durante o decurso da minha vida académica, essencialmente no curso de Engenharia Informática na Faculdade de Ciências da Universidade de Lisboa, para os vários projectos que desenvolvi para as disciplinas do curso, era sempre nos disponibilizado

vários tipos de documentação inicial para a realização destes projectos, tais como os requisitos iniciais, por vezes o modelo de dados, UML, casos de uso, entre outros, e era sempre pedido uma documentação final do projecto onde em vários destes teríamos de identificar as implementações efectuadas.

Neste departamento da Capgemini existe a falta dessa documentação, tanto de requisitos iniciais, por vezes do próprio cliente que pede alterações ou mesmo funcionalidades para uma possível aplicação, como de documentação final.

Os requisitos iniciais por vezes são levantados em reuniões com os clientes e documentados, mas são apenas requisitos gerais e não aprofundados. Estes documentos são por sua vez passados a terceiros que deverão implementar os requisitos e encontram certas dificuldades em entender o que realmente é pretendido, onde por vezes o resultado final não é o esperado pelo cliente. Os requisitos deveriam ser mais descritos e caso necessário com o apoio de casos de uso.

A documentação final é ainda mais essencial, eu principalmente senti a falta deste documento para a aplicação SigAV360, como esta aplicação já existia e como os principais elementos que desenvolveram a aplicação já não se encontravam no departamento, o documento final era essencial para o departamento.

Eu deparei-me com uma aplicação já desenvolvida onde foi necessário um maior dispêndio de tempo para entender uma funcionalidade a partir do código desenvolvido ao invés de um menor dispêndio caso existisse um documento que descrevesse os objectivos de cada uma das funcionalidades da aplicação e evitando também uma interpretação errada de algumas funcionalidades.

4.3 Trabalho futuro

Na aplicação SigAV360, apesar de ter sofrido uma reestruturação da minha parte, depois de ter participado no desenvolvimento da aplicação SigCRM, acho que ainda falta algum trabalho no SigAV360.

Com o conhecimento e experiência que adquiri com SigCRM, penso que se pode melhorar mais a aplicação. O facto de este estar desenvolvido com a Framework.NET 1.1 não ajuda muito, por isso um primeiro passo seria passar toda a aplicação para 2.0. De seguida melhoraria o sistema de notificação de mensagens da aplicação e faria algo idêntico ao SigCRM. E por fim criava controlos de pesquisa nas páginas a par da

inclusão de ordenações nas colunas das listagens, porque torna-se um pouco difícil percorrer as listas e encontrar o registo pretendido quando estas se tornam bastante numerosas.

Futuramente poder-se ia ponderar o uso de AJAX. Esta alteração seria bastante demorada, pois teria que se substituir todos os controlos das páginas para passarem a usar AJAX.

Só estas alterações melhoravam a interactividade, funcionalidade e usabilidade da aplicação.

Já na aplicação SigCRM, na minha opinião não faria grandes alterações. Esta aplicação tem as ligações certas e acessíveis, por exemplo numa entidade têm-se acesso a todos os conceitos que estão associados à entidade e as funcionalidades para cada conceito são as adequadas.

As únicas modificações que faria seriam em termos visuais da aplicação, melhoria o menu da aplicação, pois este está bastante complexo e grande. Teria que se arranjar uma forma de tornar o menu mais simples e mais visível.

Melhoria uma funcionalidade de *drag and drop* existente no conceito de processo. Este está um pouco difícil de usar e pouco funcional.

Uma vez que já é possível sincronizar tarefas com a aplicação Microsoft Outlook, já se falou na possibilidade de criar um *add-in*²⁰ do SigCRM para a aplicação Microsoft Outlook, com o apoio de algumas ferramentas Microsoft Visual Studio.

Algo que se poderia implementar nas duas aplicações seria uma funcionalidade que permitisse ao utilizador escolher as cores a serem usadas na aplicação, indo de encontro às cores da empresa.

Até à data, vai-se mesmo implementar a funcionalidade do *add-in* do SigCRM para a aplicação Microsoft Outlook e modificar o menu desta aplicação para um menu mais funcional, estilo menu da ferramenta Microsoft Office.

Ainda para a aplicação SigCRM, vão-se sempre acrescentar novas funcionalidades *standard* como estatísticas de negócio, bolsas de horas e chamadas recebidas/effectuadas

²⁰ *add-in* – Um software que estende as capacidades de uma determinada aplicação, também designado de *add-on*. Permite ser adicionado a outra aplicação.

por utilizador, etc.. E ainda terão de se implementar outras funcionalidades para os clientes que adquirem a aplicação e que pretendam acrescentar à aplicação funcionalidades de negócio específicas para eles, em que possivelmente algumas se tornaram *standard* e daí surgirá novas versões da aplicação.

Índice remissivo

A

add-in · 55

AJAX · 45

B

back-end · 20

BackLocation · 39

C

clone · 38

Code-Behind · 31

config · 36

ConnectionString · 47

cookies · 35

Cookies · 21

D

DataSet · 47

Dynamic-link library (dll) · 43

F

FreeTextBox · 49

Front-end · 8

G

GridViewExtensionPanel · 44

H

hacker · 35

L

Lista de tarefas · 11

ListPageBase · 38

LOV · 41

LOVControlBase · 41

M

Mestrado em Engenharia Informática (MEI) · 1

O

override · 37

P

PageBase · 37

PageHeader · 38

plano de testes · 33

Projecto em Engenharia Informática (PEI) · 1

R

RecordPageBase · 38

recursos · 16

recursos/resources · 32

S

Scrum · 11

SCRUM · 11; Proprietário do projecto · 11

scrum master · 12

SearchControlBase · 44

Sessions · 21

SigAV360 (Sistema Integrado de Gestão - Avaliação de Desempenho 360°) · 2, 3, 7, 14, 18

SigCRM (Sistema Integrado de Gestão – *Customer Relationship Management*) · 2, 4, 9, 15, 34
Sprint · 11

T

TexKey · 32

U

URL encoding · 39

V

variáveis de sessão · 21

VPN · 4

W

WebUserControls · 32

Z

ZedGraphWeb · 43

Bibliografia

[1] *MSDN training, Developing Microsoft ASP .Net Web Applications using Visual Studio .Net* (curso existente online em <http://www.microsoft.com/learning/syllabi/en-us/2310Bfinal.msp>)

[2] Mike Murach. *ASP.NET 2.0 web programming with C# 2005*. Mike Murach & Associates, Inc., 2005